



Open Source Used In [talos] CUA: yara-engine-go main-git-5fa594f-32

Cisco Systems, Inc.

www.cisco.com

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

Text Part Number: 78EE117C99-2772726031

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

In your requests please include the following reference number 78EE117C99-2772726031

Contents

1.1 lz4 1.9.3

1.1.1 Available under license

1.2 zlib 1.3.1

1.2.1 Available under license

1.3 zstd 1.3.1

1.3.1 Available under license

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications

1.4.2 Available under license

1.5 openssl 3.0.15

1.5.1 Available under license

1.6 curl 8.14.1-r2

1.6.1 Available under license

1.7 skalibs 2.14.4.0-r0

1.7.1 Available under license

1.8 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.8.1 Available under license

1.9 file 5.46-r2

1.9.1 Available under license

1.10 libmd 1.1.0-r0

1.10.1 Available under license

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license

1.12 libcap 2.76-r0

1.12.1 Available under license

1.13 perks 1.0.1

- 1.13.1 Available under license
- 1.14 [github.com/confluentinc/confluent-kafka-go/v2](#) 2.12.0**
 - 1.14.1 Available under license
- 1.15 [zlib](#) 1.3.1-r2**
 - 1.15.1 Available under license
- 1.16 [scanelf](#) 1.3.8-r1**
 - 1.16.1 Available under license
- 1.17 [alpine-base](#) 3.22.2-r0**
 - 1.17.1 Available under license
- 1.18 [go-yara](#) 4.3.4**
 - 1.18.1 Available under license
- 1.19 [alpine-keys](#) 2.5-r0**
 - 1.19.1 Available under license
- 1.20 [logrus](#) 1.9.3**
 - 1.20.1 Available under license
- 1.21 [yara](#) 4.5.4-r0**
 - 1.21.1 Available under license
- 1.22 [protobuf](#) 1.36.10**
 - 1.22.1 Available under license
- 1.23 [utmps](#) 0.1.3.1-r0**
 - 1.23.1 Available under license
- 1.24 [ca-certificates](#) 20250911-r0**
 - 1.24.1 Available under license
- 1.25 [libssl3](#) 3.5.4-r0**
 - 1.25.1 Available under license
- 1.26 [musl](#) 1.2.5-r10**
 - 1.26.1 Available under license
- 1.27 [aws-sdk-go-v2-internal-ini](#) 1.8.4**
 - 1.27.1 Available under license
- 1.28 [protobuf-go](#) 1.36.10**
 - 1.28.1 Available under license
- 1.29 [iputils-clockdiff](#) 20240905-r0**
 - 1.29.1 Available under license
- 1.30 [iputils-arping](#) 20240905-r0**
 - 1.30.1 Available under license
- 1.31 [apk-tools](#) 2.14.9-r3**
 - 1.31.1 Available under license
- 1.32 [kelseyhightower-envconfig](#) 1.4.0**
 - 1.32.1 Available under license

1.33 go.yaml.in/yaml/v2 2.4.3

1.33.1 Available under license

1.34 openssl 3.5.4-r0

1.34.1 Available under license

1.35 bash 5.2.37-r0

1.35.1 Available under license

1.36 go.yaml.in-yaml 2.4.3

1.36.1 Available under license

1.37 pam 1.7.0-r4

1.37.1 Available under license

1.38 readline 8.2.13-r1

1.38.1 Available under license

1.39 ca-certificates-bundle 20250911-r0

1.39.1 Available under license

1.40 busybox 1.37.0-r19

1.40.1 Available under license

1.41 shadow 4.17.3-r0

1.41.1 Available under license

1.42 libncursesw 6.5_p20250503-r0

1.42.1 Available under license

1.43 ncurses 6.5_p20250503-r0

1.43.1 Available under license

1.44 utmps-lib 0.1.3.1-r0

1.44.1 Available under license

1.45 iputils-ping 20240905-r0

1.45.1 Available under license

1.46 libmagic 5.46-r2

1.46.1 Available under license

1.47 libcrypto3 3.5.4-r0

1.47.1 Available under license

1.48 alpine-baselayout-data 3.7.0-r0

1.48.1 Available under license

1.49 github.com/hillu/go-yara/v4 4.3.4

1.49.1 Available under license

1.50 iputils-tracepath 20240905-r0

1.50.1 Available under license

1.51 iputils 20240905-r0

1.51.1 Available under license

1.52 alpine-baselayout 3.7.0-r0

- 1.52.1 Available under license
- 1.53 alpine-release 3.22.2-r0**
 - 1.53.1 Available under license
- 1.54 libbsd 0.12.2-r0**
 - 1.54.1 Available under license
- 1.55 confluentinc-confluent-kafka-go 2.12.0**
 - 1.55.1 Available under license
- 1.56 ssl-client 1.37.0-r19**
 - 1.56.1 Available under license
- 1.57 libapk2 2.14.9-r3**
 - 1.57.1 Available under license
- 1.58 clientmodel 0.6.2**
 - 1.58.1 Available under license
- 1.59 skalibs-libs 2.14.4.0-r0**
 - 1.59.1 Available under license
- 1.60 musl-utils 1.2.5-r10**
 - 1.60.1 Available under license
- 1.61 github.com/aws/aws-sdk-go-v2/internal/ini 1.8.4**
 - 1.61.1 Available under license
- 1.62 sirupsen-logrus 1.9.3**
 - 1.62.1 Available under license
- 1.63 cJSON 1.7.19-r0**
 - 1.63.1 Notifications
 - 1.63.2 Available under license
- 1.64 prometheus-client-model 0.6.2**
 - 1.64.1 Available under license
- 1.65 prometheus-client 1.23.2**
 - 1.65.1 Available under license
- 1.66 xxhash 2.3.0**
 - 1.66.1 Available under license
- 1.67 procfs 0.19.2**
 - 1.67.1 Available under license
- 1.68 stdlib go1.25.4**
 - 1.68.1 Available under license
- 1.69 golang 1.25.4**
 - 1.69.1 Available under license
- 1.70 github.com/aws/aws-sdk-go-v2/aws/protocol/eventstream 1.7.3**
 - 1.70.1 Available under license
- 1.71 aws-sdk-go-v2-service-internal-accept-encoding 1.13.3**

- 1.71.1 Available under license
- 1.72 [github.com/aws/smithy-go](#) 1.23.2**
 - 1.72.1 Available under license
- 1.73 [aws-sdk-go-v2-aws-protocol-eventstream](#) 1.7.3**
 - 1.73.1 Available under license
- 1.74 [smithy-go](#) 1.23.2**
 - 1.74.1 Available under license
- 1.75 [github.com/aws/aws-sdk-go-v2/service/internal/accept-encoding](#) 1.13.3**
 - 1.75.1 Available under license
- 1.76 [x-sys](#) 0.38.0**
 - 1.76.1 Available under license
- 1.77 [common](#) 0.67.3**
 - 1.77.1 Available under license
- 1.78 [prometheus-common](#) 0.67.3**
 - 1.78.1 Available under license
- 1.79 [aws-sdk-go-v2-service-s3](#) 1.91.1**
 - 1.79.1 Available under license
- 1.80 [github.com/aws/aws-sdk-go-v2/service/internal/checksum](#) 1.9.5**
 - 1.80.1 Available under license
- 1.81 [github.com/aws/aws-sdk-go-v2/credentials](#) 1.19.0**
 - 1.81.1 Available under license
- 1.82 [github.com/aws/aws-sdk-go-v2/internal/configsources](#) 1.4.14**
 - 1.82.1 Available under license
- 1.83 [github.com/aws/aws-sdk-go-v2/internal/endpoints/v2](#) 2.7.14**
 - 1.83.1 Available under license
- 1.84 [aws-sdk-go-v2-service-internal-checksum](#) 1.9.5**
 - 1.84.1 Available under license
- 1.85 [github.com/aws/aws-sdk-go-v2/service/internal/s3shared](#) 1.19.14**
 - 1.85.1 Available under license
- 1.86 [github.com/aws/aws-sdk-go-v2/feature/s3/manager](#) 1.20.9**
 - 1.86.1 Available under license
- 1.87 [aws-sdk-go-v2-feature-s3-manager](#) 1.20.9**
 - 1.87.1 Available under license
- 1.88 [aws-sdk-go-v2-internal-configsources](#) 1.4.14**
 - 1.88.1 Available under license
- 1.89 [github.com/aws/aws-sdk-go-v2/config](#) 1.32.0**
 - 1.89.1 Available under license
- 1.90 [github.com/aws/aws-sdk-go-v2/internal/v4a](#) 1.4.14**
 - 1.90.1 Available under license

- 1.91 aws-sdk-go-v2-service-ssooidc 1.35.8**
 - 1.91.1 Available under license
- 1.92 aws-sdk-go-v2-credentials 1.19.0**
 - 1.92.1 Available under license
- 1.93 aws-sdk-go-v2-feature-ec2-imds 1.18.14**
 - 1.93.1 Available under license
- 1.94 aws-sdk-go-v2 1.40.0**
 - 1.94.1 Available under license
- 1.95 aws-aws-sdk-go-v2-service-signin 1.0.1**
 - 1.95.1 Available under license
- 1.96 aws-sdk-go-v2-config 1.32.0**
 - 1.96.1 Available under license
- 1.97 github.com/aws/aws-sdk-go-v2/service/signin 1.0.1**
 - 1.97.1 Available under license
- 1.98 aws-sdk-go-v2-service-sso 1.30.4**
 - 1.98.1 Available under license
- 1.99 aws-sdk-go-v2-service-internal-s3shared 1.19.14**
 - 1.99.1 Available under license
- 1.100 github.com/aws/aws-sdk-go-v2/service/ssooidc 1.35.8**
 - 1.100.1 Available under license
- 1.101 aws-sdk-go-v2-internal-endpoints 2.7.14**
 - 1.101.1 Available under license
- 1.102 github.com/aws/aws-sdk-go-v2/service/internal/presigned-url 1.13.14**
 - 1.102.1 Available under license
- 1.103 github.com/aws/aws-sdk-go-v2/service/sso 1.30.4**
 - 1.103.1 Available under license
- 1.104 github.com/aws/aws-sdk-go-v2/service/sts 1.41.1**
 - 1.104.1 Available under license
- 1.105 aws-sdk-go-v2-service-sts 1.41.1**
 - 1.105.1 Available under license
- 1.106 github.com/aws/aws-sdk-go-v2/feature/ec2/imds 1.18.14**
 - 1.106.1 Available under license
- 1.107 github.com/aws/aws-sdk-go-v2/service/s3 1.91.1**
 - 1.107.1 Available under license
- 1.108 aws-sdk-go-v2-service-internal-presigned-url 1.13.14**
 - 1.108.1 Available under license
- 1.109 aws-sdk-go-v2-internal-v4a 1.4.14**
 - 1.109.1 Available under license

1.1 lz4 1.9.3

1.1.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

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

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

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

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of

Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

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

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

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

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or

distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

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

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

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

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

This section is intended to make thoroughly clear what is believed to

be a consequence of the rest of this License.

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

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

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

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

NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,

INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

it

under certain conditions; type ``show c'` for details.

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

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

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

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General
Public License instead of this License.

Copyright (c) 2014, lpsantil

All rights reserved.

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

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

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

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

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

This repository uses 2 different licenses :

- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPLv2 license, unless explicitly stated otherwise

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

This model is selected to emphasize that
files in the `lib` directory are designed to be included into 3rd party applications,
while all other files, in `programs`, `tests` or `examples`,
receive more limited attention and support for such scenario.

LZ4 Library

Copyright (c) 2011-2016, Yann Collet

All rights reserved.

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

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

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

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

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

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: liblz4

Upstream-Contact: Yann Collet <Cyan4973@github.com>

Source: <https://github.com/lz4/lz4>

Files: *

Copyright: (C) 2011-2020 Yann Collet

License: GPL-2+

The full text of license: <https://github.com/lz4/lz4/blob/dev/lib/LICENSE>

1.2 zlib 1.3.1

1.2.1 Available under license :

Copyright notice:

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

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

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

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

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

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

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

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

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

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

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <http://tools.ietf.org/html/rfc1950> (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).

*/

Boost Software License - Version 1.0 - August 17th, 2003

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/* gzip.c -- zlib functions common to reading and writing gzip files

* Copyright (C) 2004-2024 Mark Adler

* For conditions of distribution and use, see copyright notice in zlib.h

*/

1.3 zstd 1.3.1

1.3.1 Available under license :

BSD-style without advertising clause

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
(<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.4.2 Available under license :

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

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

AND FITNESS FOR A PARTICULAR

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

*/

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

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

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

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

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

PERFORMANCE OF THIS SOFTWARE.

By:

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

Share and Enjoy! :-)

regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

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

"

FindZstd.cmake: [git@github.com:facebook/folly](https://github.com/facebook/folly).git 87f1a403b49552dae75ae94c8610dd5979913477

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

and conversions to other media types.

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

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

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

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

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

You may add Your own copyright statement to Your modifications

and

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for

reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed

to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if

You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

For src/crc32c.c copied (with modifications) from

<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction

* Copyright (C) 2013 Mark Adler

* Version 1.1 1 Aug 2013 Mark Adler

*/

/*

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

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

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

Mark Adler

madler@alumni.caltech.edu

*/

For sys/queue.h:

* Copyright (c) 1991, 1993

* The Regents of the University of California. All rights reserved.

*

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

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

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

For cJSON.c and cJSON.h:

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

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

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

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

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.

All rights reserved.

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

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

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

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

LICENSE.cjson

For cJSON.c and cJSON.h:

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

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

to do so, subject to the following conditions:

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

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

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

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

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

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

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

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

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a

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

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

LONDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,

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

By:

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

Share and Enjoy! :-)

LICENSE.hdrhistogram

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

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

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

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

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

LICENSE.lz4

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

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

LICENSE.murmur2

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

MurMurHash2 Library

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

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

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

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

LICENSE.nanopb

For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

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

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

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

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

LICENSE.opentelemetry

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

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

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

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

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

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

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

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

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

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

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

incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

origin of the Work and reproducing the content of the NOTICE file.

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

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

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

LICENSE.pycrc

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

=====

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

Permission is hereby granted,

free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

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

LICENSE.queue

For

sys/queue.h:

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

LICENSE.regexp

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

"

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

"

LICENSE.snappy

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h

```
# originally retrieved from http://github.com/andikleen/snappy-c #
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####
```

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

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

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

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

LICENSE.tinythread

From <https://github.com/tinythread/tinythread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

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

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

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

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

LICENSE.wingetopt

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

```
/*
 * Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
 * ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Sponsored in part by the Defense Advanced Research Projects
 * Agency (DARPA) and Air Force Research Laboratory, Air Force
 * Materiel Command, USAF, under agreement number F39502-99-1-0512.
 */
/*-
 * Copyright (c) 2000 The NetBSD Foundation, Inc.
 * All rights reserved.
 */
```

* This code is derived from software contributed to The NetBSD Foundation
 * by Dieter Baron and Thomas Klausner.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice,
 * this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
 #####
 # LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
 # originally retrieved from http://github.com/andikleen/snappy-c #
 # git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
 #####

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

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

this software without specific prior written permission.

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

librdkafka - Apache Kafka C driver library

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

All rights reserved.

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

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

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

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

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

.

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

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

Files: *

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

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

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

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

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

Files: src/snappy.c src/snappy.h src/snappy_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

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

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

Files: src/rdxxhash.h src/rdxxhash.c

Copyright: 2012-2014, Yann Collet

License: BSD-2-clause

Files: src/regex.h src/regex.c

Copyright: n/a

License: public-domain

License: public-domain

The files tagged with this license contain the following paragraphs:

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

Files: debian/*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

BSD-style without advertising clause

Copyright (c) 2016-2019, Magnus Edenhill
All rights reserved.

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

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

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

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

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

For files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

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

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

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

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

distribution.

parts of src/rdmurmur2.c: [git@github.com:abandoned/murmur2.git](https://github.com:abandoned/murmur2.git)

MurMurHash2 Library

//-----

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

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

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

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

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

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

.

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

.

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

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

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

Files: *

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

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

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

.

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

.

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

Files: src/snappy.c src/snappy.h src/snappy_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

.

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

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

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

.

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

Files: debian/*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

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

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

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

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

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

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

License

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

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

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

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

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

=====

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

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

1.5 openssl 3.0.15

1.5.1 Available under license :

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

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

Terms of the Perl programming language system itself

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

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

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

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

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

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

Preamble

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

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free

software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

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

- a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and
- b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.
- d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

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

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

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

corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

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

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

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

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

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

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

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

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

Foundation.

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

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

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

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

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

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

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

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

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

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

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

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

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

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

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

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

Definitions:

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

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived

from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

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

- a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
- b) use the modified Package only within your corporation or organization.
- c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
- b) accompany the distribution with the machine-readable source of the Package with your modifications.
- c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

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

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

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

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

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

The End

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this

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

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

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

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

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

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

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

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

END OF TERMS AND CONDITIONS

1.6 curl 8.14.1-r2

1.6.1 Available under license :

Copyright (c) <year> <owner>.

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

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

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

COPYRIGHT AND PERMISSION NOTICE

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

All rights reserved.

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

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this

Software without prior written authorization of the copyright holder.
BSD-4-Clause (University of California-Specific)

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

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

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

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

COPYRIGHT AND PERMISSION NOTICE

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

All rights reserved.

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

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

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

1.7 skalibs 2.14.4.0-r0

1.7.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

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

Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in

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

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

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

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

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

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

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

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

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

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

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

6.

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

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

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

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

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

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

the limitation as if written in the body of this License.

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

The hypothetical commands

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

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

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

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildid="$srcdir/${_pkgreal}-$pkgver"
```

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

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

```

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

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

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

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

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
    bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0

```


11

11

```

--prefix=/usr
make
}

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

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

```

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

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

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

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

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

```

subpackages="$pkgname-list"

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

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

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

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

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

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

```

```

builddir="$srcdir"
options="!check" # just headers

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

```

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

Main Xfig copyright notice:

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

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

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

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

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
 The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
 "X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>
 # Contributor: Dhruvin Gandhi <contact@dhruvin>
 # Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
 pkgname=py3-license-expression

```

_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

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

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

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

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright

```

- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 3. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
- BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * @(#)queue.h 8.5 (Berkeley) 8/20/94
- */

```
#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_
```

```
/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The
 * elements are singly linked for minimum space and
 * pointer manipulation
 * overhead at the expense of O(n) removal for arbitrary elements. New
 * elements can be added to the list after an existing element or at the
 * head of the list. Elements being removed from the head of the list
 * should use the explicit macro for this purpose for optimum
 * efficiency. A singly-linked list may only be traversed in the forward
 * direction. Singly-linked lists are ideal for applications with large
 * datasets and few or no removals or for implementing a LIFO queue.
 *
 * A list is headed by a single forward pointer (or an array of forward
 * pointers for a hash table header). The elements are doubly linked
 * so that an arbitrary element can be removed without a need to
 * traverse the list. New elements can be added to the list before
 * or after an existing element or at the head of the list. A list
 * may only be traversed in the forward direction.
 *
 * A simple queue is headed by a pair of pointers, one the head of the
```

- * list and the other to the tail of the list. The elements are singly linked to save space, so elements can only be removed from the head of the list. New elements can be added to the list after an existing element, at the head of the list, or at the end of the list. A simple queue may only be traversed in the forward direction.
- *
- * A tail queue is headed by a pair of pointers, one to the head of the list and the other to the tail of the list. The elements are doubly linked so that an arbitrary element can be removed without a need to traverse the list. New elements can be added to the list before or after an existing element, at the head of the list, or at the end of the list. A tail queue may be traversed in either direction.
- *
- * A circle queue is headed by a pair of pointers, one to the head of the list and the other to the tail of the list. The elements are doubly linked so that an arbitrary element can be removed without a need to traverse the list. New elements can be added to the list before or after an existing element, at the head of the list, or at the end of the list.
- * A circle queue may be traversed in either direction, but has a more complex end of list detection.
- *
- * For details on the use of these macros, see the queue(3) manual page.
- */

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

```
#ifdef __NetBSD__
#include <sys/null.h>
#endif
```

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

```
/*
 * Singly-linked List definitions.
```

```

*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

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

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

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

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

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

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

#define SLIST_INSERT_HEAD(head, elm, field) do { \

```



```

(elm)->field.sle_next = (head)->slh_first; \
(head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)

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

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

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

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

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

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

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)

```

```

#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

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

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

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

/*
 * List functions.
 */

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

```

```

field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

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

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

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

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

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

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \

```

```

(elm2)->field.le_prev = (elm)->field.le_prev; \
*(elm2)->field.le_prev = (elm2); \
QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

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

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

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

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

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

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head) && \
     ((next = ((var)->field.sqe_next)), 1); \
     (var) = (next))

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

```

```

= &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

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

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

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

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

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

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

#define SIMPLEQ_CONCAT(head1, head2) do { \
if (!SIMPLEQ_EMPTY((head2))) { \

```

```

*(head1)->sqh_last = (head2)->sqh_first; \
(head1)->sqh_last = (head2)->sqh_last; \
SIMPLEQ_INIT((head2)); \
} \
} while (/*CONSTCOND*/0)

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

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

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

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

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
(*(((struct headname *) (void *)((head)->tqh_last))->tqh_last))
#define TAILQ_PREV(elm, headname, field) \
(*(((struct headname \
*)(void *)((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \

```

```

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

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

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

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

/*
 * Tail queue functions.
 */
#if defined(QEUEDEBBUG)
#define QEUEDEBBUG_TAILQ_INSERT_HEAD(head,
elm, field) \
if ((head)->tqh_first && \
    (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
    QEUEDEBBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QEUEDEBBUG_TAILQ_INSERT_TAIL(head, elm, field) \
if (*(head)->tqh_last != NULL) \
    QEUEDEBBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QEUEDEBBUG_TAILQ_OP(elm, field) \
if ((elm)->field.tqe_next && \
    (elm)->field.tqe_next->field.tqe_prev != \
    &(elm)->field.tqe_next) \
    QEUEDEBBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
        __FILE__, __LINE__); \
if (*(elm)->field.tqe_prev != (elm)) \
    QEUEDEBBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
        __FILE__, __LINE__);
#define QEUEDEBBUG_TAILQ_PREREMOVE(head, elm, field) \
if ((elm)->field.tqe_next == NULL && \
    (head)->tqh_last != &(elm)->field.tqe_next) \
    QEUEDEBBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
        (head), (elm), __FILE__,

```

```

__LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

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

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

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

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

```



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

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

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

```
#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)
```

```
/*
```

```
* Singly-linked Tail queue declarations.
```

```
*/
```

```
#define STAILQ_HEAD(name, type) \
```

```

struct name {      \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

#ifdef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros

```

```

* for compatibility but because of pointer aliasing issues their use
* is discouraged!
*/

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

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

```

```

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

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

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

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

/*
 * Circular queue functions.
 */
#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \

```

```

(head)->cqh_last = (elm); \
else \
(listelm)->field.cqe_next->field.cqe_prev = (elm); \
(listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

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

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

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

#define CIRCLEQ_REMOVE(head, elm, field) do { \
QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (elm), field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
(head)->cqh_last = (elm)->field.cqe_prev; \
else \

```

```

(elm)->field.cqe_next->field.cqe_prev = \
    (elm)->field.cqe_prev; \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
    (head)->cqh_first = (elm)->field.cqe_next; \
else \
    (elm)->field.cqe_prev->field.cqe_next = \
        (elm)->field.cqe_next; \
QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

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

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

/*
 * Circular queue access methods.
 */

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

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

#endif /* !_SYS_QUEUE_H_ */

```

```

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

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

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

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

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

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

#undef __P
#undef __PMT

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

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

```



```

#define __STRING(x) #x

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

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

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

prepare() {
    default_prepare

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

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

check() {
    cargo test --frozen
}

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

```

```

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

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

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

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

```

```

#endif

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

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

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

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

```

```

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

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

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

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

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

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

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
    void name##_SPLAY(struct name *, struct type *); \
    void name##_SPLAY_MINMAX(struct name *, int); \
    struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
    struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
    \
/* Finds the node with the same key as elm */ \

```

```

static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if(__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\

```

```

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

```

```

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

```

```

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

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

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

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

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

#define RB_INITIALIZER(root) \
{ \
    NULL }

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

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \

```



```

struct type *rbe_left; /* left element */ \
struct type *rbe_right; /* right element */ \
struct type *rbe_parent; /* parent element */ \
int rbe_color; /* node color */ \
}

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

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

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

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

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

```

```

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

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

/* Main rb operation.
 * Moves node close to the key of elm to top
 */
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \

```

```

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

```

```

{
    \
    struct type *tmp;
    \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) {
        \
        if (RB_LEFT(parent, field) == elm) {
            \
            tmp = RB_RIGHT(parent, field);
            \
            if (RB_COLOR(tmp, field) == RB_RED) {
                \
                RB_SET_BLACKRED(tmp, parent, field);
                \
                RB_ROTATE_LEFT(head, parent, tmp, field);
                \
                tmp = RB_RIGHT(parent, field);
            }
            \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp,
field), field) == RB_BLACK)) {
                \
                RB_COLOR(tmp, field) = RB_RED;
                \
                elm = parent;
                \
                parent = RB_PARENT(elm, field);
            } else {
                \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {
                    \
                    struct type *oleft;
                    \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL)
                        \
                        RB_COLOR(oleft, field) = RB_BLACK;
                    \
                    RB_COLOR(tmp, field) = RB_RED;
                    \
                    RB_ROTATE_RIGHT(head, tmp, oleft, field);
                    \
                    tmp = RB_RIGHT(parent, field);
                }
                \
                RB_COLOR(tmp, field) = RB_COLOR(parent, field);
                \
                RB_COLOR(parent, field) = RB_BLACK;
                \
                if (RB_RIGHT(tmp, field))
                    \
                    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;
                \
                RB_ROTATE_LEFT(head, parent, tmp, field);
                \
                elm = RB_ROOT(head);
                \
                break;
            }
            \
        } else {
            \
            tmp = RB_LEFT(parent, field);
            \
            if (RB_COLOR(tmp, field) == RB_RED) {
                \
                RB_SET_BLACKRED(tmp, parent, field);
                \
                RB_ROTATE_RIGHT(head, parent, tmp, field);
                \
                tmp
= RB_LEFT(parent, field);
            }
            \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,

```

```

y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

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

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

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

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

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

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

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"

```

source="https://kernel.org/pub/linux/kernel/v6.x/linux-\$_kernver.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

1.8 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.8.1 Available under license :

Copyright (c) 2011, Open Knowledge Foundation Ltd.
All rights reserved.

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

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

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

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

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

1.9 file 5.46-r2

1.9.1 Available under license :

\$File: COPYING,v 1.2 2018/09/09 20:33:28 christos Exp \$
Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995.
Software written by Ian F. Darwin and others;
maintained 1994- Christos Zoulas.

This software is not subject to any export provision of the United States Department of Commerce, and may be exported to any country or planet.

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

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

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

1.10 libmd 1.1.0-r0

1.10.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

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

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

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

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

Files:

include/sha2.h

src/sha2.c

Copyright:

Copyright 2000-2001, Aaron D. Gifford

All rights reserved.

License: BSD-3-clause-Aaron-D-Gifford

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

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

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

OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Files:

include/rmd160.h

src/rmd160.c

Copyright:

Copyright 2001 Markus Friedl. All rights reserved.

License: BSD-2-clause

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

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

.

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

Files:

src/md2.c

Copyright:

Copyright (c) 2001 The NetBSD Foundation, Inc.

All rights reserved.

.

This code is derived from software contributed to The NetBSD Foundation
by Andrew Brown.

License: BSD-2-clause-NetBSD

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

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must
reproduce the above copyright
notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

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

Files:

man/rmd160.3

man/sha1.3

man/sha2.3

Copyright:

Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>

License: ISC

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

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

Files:

man/mdX.3

src/helper.c

Copyright:

Poul-Henning Kamp <phk@login.dkuug.dk>

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this file. As long as you retain
this notice you

can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

Files:

include/md4.h

src/md4.c

Copyright:
Colin Plumb
Todd C. Miller
License: public-domain-md4
This code implements the MD4 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.
Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:
include/md5.h
src/md5.c
Copyright:
Colin Plumb
License: public-domain-md5
This code implements the MD5 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.

Files:
include/sha1.h
src/sha1.c
Copyright:
Steve Reid <steve@edmweb.com>
License: public-domain-sha1
100% Public
Domain

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

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

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

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

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you

received the program
in object code or executable form with such
an offer, in accord with Subsection b above.)

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

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

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

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

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

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

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

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

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

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

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

Each version is given a distinguishing version number. If the Program

specifies

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

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

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

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

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

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

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

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

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

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

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.12 libcap 2.76-r0

1.12.1 Available under license :

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/cap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

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

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you

have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another

language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities

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

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

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

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

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

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

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

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

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

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

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

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a

special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

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

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

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

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

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

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

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

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

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

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

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

10. If you wish to incorporate parts of the Program into other free

programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

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

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

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

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

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

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

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

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

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

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

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

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

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

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/psx release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

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

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the

terms of the
GNU General Public License (v2.0 - see below), in which case the
provisions of the GNU GPL are required INSTEAD OF the above
restrictions. (This clause is necessary due to a potential conflict
between the GNU GPL and the restrictions contained in a BSD-style
copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.

This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

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

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

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

source code. And you must show them these terms so they know their rights.

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

Activities

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

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

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

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

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

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

collective works based on the Program.

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

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

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

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

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

4. You may not copy, modify, sublicense, or distribute the Program

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

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

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

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

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

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

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

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

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

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

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

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

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

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

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

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

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

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

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

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

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

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

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the
licensed conditions under which the contents of this libcap release
may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap, with

or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name
of any author may not be used to endorse or promote
products derived from this software without their specific prior
written permission.

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the
GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

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

Preamble

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

it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

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

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

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

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

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

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free

program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

In addition,

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

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

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

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

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

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

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

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

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

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

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

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

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

the limitation as if written in the body of this License.

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if

necessary. Here is a sample; alter the names:

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

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public

License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

/* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later */

Unless otherwise **explicitly** stated the following text describes the
licensed conditions under which the contents of this module release
may be distributed:

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

1. Redistributions of source code must retain any existing copyright
notice, and this entire permission notice in its entirety,
including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current
copyright notices, this list of conditions, and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
3. The name of any author may not be used to endorse or promote
products derived from this software without their specific prior
written
permission.

ALTERNATIVELY, this product may be distributed under the terms of the
GNU Library General Public License, in which case the provisions of
the GNU LGPL are required INSTEAD OF the above restrictions. (This
clause is necessary due to a potential conflict between the GNU LGPL
and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

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

1.13 perks 1.0.1

1.13.1 Available under license :

Copyright (C) 2013 Blake Mizerany

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

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

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

1.14 github.com/confluentinc/confluent-kafka-go/v2 2.12.0

1.14.1 Available under license :

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.
All rights reserved.

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

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

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

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

LICENSE.cjson

For cJSON.c and cJSON.h:

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

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

to do so, subject to the following conditions:

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

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

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

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

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

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

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

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

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a

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

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

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO

EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> /oo/

<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

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

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

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

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

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

LICENSE.lz4

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

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

LICENSE.murmur2

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

MurMurHash2 Library

//-----

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

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

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

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

LICENSE.nanopb

For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

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

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

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

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

LICENSE.opentelemetry

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

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

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

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

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

incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

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

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

LICENSE.pycrc

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

=====

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

Permission is hereby granted,

free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

LICENSE.queue

For

sys/queue.h:

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

LICENSE.regexp

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

"

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

"

LICENSE.snappy

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h

```
# originally retrieved from http://github.com/andikleen/snappy-c #
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####
```

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

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

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

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

LICENSE.tinythread

From <https://github.com/tinythread/tinythread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

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

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

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

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

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

```
/*
 * Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
 * ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Sponsored in part by the Defense Advanced Research Projects
 * Agency (DARPA) and Air Force Research Laboratory, Air Force
 * Materiel Command, USAF, under agreement number F39502-99-1-0512.
 */
/*-
 * Copyright (c) 2000 The NetBSD Foundation, Inc.
 * All rights reserved.
 */
```


* This code is derived from software contributed to The NetBSD Foundation
 * by Dieter Baron and Thomas Klausner.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice,
 this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
 and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
 the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
 other entities that control, are controlled by, or are under common
 control with that entity. For the purposes of this definition,
 "control" means (i) the power, direct or indirect, to cause the
 direction or management of such entity, whether by contract or
 otherwise, or (ii) ownership of fifty percent (50%) or more of the
 outstanding shares, or (iii) beneficial ownership of such entity.

"You"

(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.15 zlib 1.3.1-r2

1.15.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.16 scanelf 1.3.8-r1

1.16.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission

to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program
or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program
(or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program
in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled
to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under

this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing

to distribute
software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies
a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED
BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this

when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical
commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.17 alpine-base 3.22.2-r0

1.17.1 Available under license :

MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT

1.18 go-yara 4.3.4

1.18.1 Available under license :

Copyright (c) 2015-2020 Hilko Bengen <bengen@hilluzination.de>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.19 alpine-keys 2.5-r0

1.19.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

OR OTHER DEALINGS IN THE SOFTWARE.

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=alpine-keys

pkgver=2.5

pkgrel=0

pkgdesc="Public keys for Alpine Linux packages"

url="https://alpinelinux.org"

we install arch specific keys to /etc so we cannot do arch=noarch

arch="all"

license="MIT"

replaces="alpine-base"

1.20 logrus 1.9.3

1.20.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.21 yara 4.5.4-r0

1.21.1 Available under license :

```
# This is the official list of people who can contribute
# (and typically have contributed) code to the YARA repository.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
#   http://code.google.com/legal/individual-cla-v1.0.html
#   http://code.google.com/legal/corporate-cla-v1.0.html
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate
# CLA was used.

# Names should be added to this file like so:
#   Name <email address>

# Please keep the list sorted.

Anthony Desnos <adesnos@google.com>
Antonio Vargas Gonzalez <winden@google.com>
Christian Blichmann <cblichmann@google.com>
Hilko Bengen <bengen@hilluzination.de>
Joachim Metz <j Joachim.metz@gmail.com>
Karel Hajek <houndthe@protonmail.com>
Karl Hiramoto <karl.hiramoto@virustotal.com>
Mike Wiacek <mjwiacek@google.com>
Shane Huntley <shuntley@google.com>
Stefan Buehlmann <stefan.buehlmann@joebox.org>
Victor M. Alvarez <plusvic@gmail.com>;<vmalvarez@virustotal.com>
Wesley Shields <wxs@atarininja.org>

Copyright (c) 2007-2016. The YARA Authors. All Rights Reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.22 protobuf 1.36.10

1.22.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.23 utmps 0.1.3.1-r0

1.23.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim

copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not

compelled to copy
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands

`show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgrel is used by apkgbuild-cpan to find modules at MetaCpan
_pkgrel=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
```

```

license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {

```

```

# some GNU autoconf options are ignored, but it works
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--sourcedir=/usr/share/$pkgname
make
}

package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21

```

```

pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
    update_config_sub
    default_prepare
}

build() {
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr
    make
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

subpackages="\$pkgname-list"

_types="html json rdfa rdfnt rdfturtle rdfxml template text"

for type in \$_types; do

subpackages="\$subpackages \$pkgname-\$type:_subpkg"

done

package() {

mkdir -p "\$pkgdir"

}

_subpkg() {

local type=\${subpkgname/\$pkgname-/}

pkgdesc="\$pkgdesc (\$type)"

install_if="\$pkgname=\$pkgver-r\$pkgrel"

mkdir -p "\$subpkgdir"/usr/share/spdx

cp -r "\$builddir"/\$type "\$subpkgdir"/usr/share/spdx/

}

list() {

pkgdesc="\$pkgdesc (licence list)"

mkdir -p "\$subpkgdir"/usr/share/spdx

local i; for i in "\$builddir"/text/*.txt;

do

local license=\${i##*/}

echo \${license%.*} >> "\$subpkgdir"/usr/share/spdx/license.lst

```

done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}

sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

```

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement

being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

Contributor: Dhruvin Gandhi <contact@dhruvin>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

_pkgname=license_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\${_pkgname}/\${_pkgname}-\${pkgver}.tar.gz"

builddir="\$srcdir/\${_pkgname}-\${pkgver}"

build() {

gpep517 build-wheel \

--wheel-dir .dist \

--output-fd 3 3>&1 >&2

}

check() {

python3 -m venv --clear --without-pip --system-site-packages .testenv

.testenv/bin/python3 -m installer .dist/*.whl

.testenv/bin/python3 -m pytest

}

package() {


```

python3 -m installer -d "$pkgdir" \
    .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 * BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)queue.h 8.5 (Berkeley) 8/20/94
 */

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
 * This file defines five types of data structures: singly-linked lists,

```

- * lists, simple queues, tail queues, and circular queues.
- *
- * A singly-linked list is headed by a single forward pointer. The
- * elements are singly linked for minimum space and
- pointer manipulation
- * overhead at the expense of $O(n)$ removal for arbitrary elements. New
- * elements can be added to the list after an existing element or at the
- * head of the list. Elements being removed from the head of the list
- * should use the explicit macro for this purpose for optimum
- * efficiency. A singly-linked list may only be traversed in the forward
- * direction. Singly-linked lists are ideal for applications with large
- * datasets and few or no removals or for implementing a LIFO queue.
- *
- * A list is headed by a single forward pointer (or an array of forward
- * pointers for a hash table header). The elements are doubly linked
- * so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before
- * or after an existing element or at the head of the list. A list
- * may only be traversed in the forward direction.
- *
- * A simple queue is headed by a pair of pointers, one the head of the
- * list and the
- other to the tail of the list. The elements are singly
- * linked to save space, so elements can only be removed from the
- * head of the list. New elements can be added to the list after
- * an existing element, at the head of the list, or at the end of the
- * list. A simple queue may only be traversed in the forward direction.
- *
- * A tail queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before or
- * after an existing element, at the head of the list, or at the end of
- * the list. A tail queue may be traversed in either direction.
- *
- * A circle queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements
- can be added to the list before or after
- * an existing element, at the head of the list, or at the end of the list.
- * A circle queue may be traversed in either direction, but has a more
- * complex end of list detection.
- *
- * For details on the use of these macros, see the queue(3) manual page.
- */
- /*

```

* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere. This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDDEBUG)
# if defined(_KERNEL)
#  define QUEUEDDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

```

```

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \
    (var) != SLIST_END(head) && \
    ((tvar) = SLIST_NEXT((var), field), 1); \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do { \
    (elm)->field.sle_next = (slistelm)->field.sle_next; \
    (slistelm)->field.sle_next \
    = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do { \
    (elm)->field.sle_next = (head)->slh_first; \
    (head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do { \
    (slistelm)->field.sle_next = \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)

```

```

/*
 * List definitions.
 */
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
     (var) != LIST_END(head) && \
     ((tvar) = LIST_NEXT((var), field), 1); \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */

```

```

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field) \
if ((head)->lh_first && \
    (head)->lh_first->field.le_prev != &(head)->lh_first) \
    QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
if ((elm)->field.le_next && \
    (elm)->field.le_next->field.le_prev != \
    &(elm)->field.le_next) \
    QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
        __FILE__, __LINE__); \
if (*(elm)->field.le_prev != (elm)) \
    QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
(elm)->field.le_next = (void *)1L; \
(elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm, field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \

```

```

QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
if (((elm)->field.le_next
= (head)->lh_first) != LIST_END(head))\
    (head)->lh_first->field.le_prev = &(elm)->field.le_next;\
(head)->lh_first = (elm); \
(elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

```

```

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

```

```

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \
    (elm2)->field.le_prev = (elm)->field.le_prev; \
    *(elm2)->field.le_prev = (elm2); \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

```

```

/*

```

```

* Simple queue definitions.

```

```

*/

```

```

#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

```

```

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

```

```

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

```

```

/*

```

```

* Simple queue access methods.

```

```

*/

```

```

#define SIMPLEQ_FIRST(head) ((head)->sqh_first)

```

```

#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->sqh_first); \
        (var) != SIMPLEQ_END(head); \
        (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->sqh_first); \
        (var) != SIMPLEQ_END(head) && \
        ((next = ((var)->field.sqe_next)), 1); \
        (var) = (next))

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last \
    = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \

```



```

if
(((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
 == NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
if ((head)->sqh_first == (elm)) { \
SIMPLEQ_REMOVE_HEAD((head), field); \
} else { \
struct type *curelm = (head)->sqh_first; \
while (curelm->field.sqe_next != (elm)) \
curelm = curelm->field.sqe_next; \
if ((curelm->field.sqe_next = \
curelm->field.sqe_next->field.sqe_next) == NULL) \
(head)->sqh_last = &(curelm)->field.sqe_next; \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
if (!SIMPLEQ_EMPTY((head2))) { \
*(head1)->sqh_last = (head2)->sqh_first; \
(head1)->sqh_last = (head2)->sqh_last; \
SIMPLEQ_INIT((head2)); \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
(SIMPLEQ_EMPTY((head)) ? \
NULL : \
((struct type \
*)(void *) \
((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
struct name { \
qual type *tqh_first; /* first element */ \
qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \

```

```

struct {
    \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *) ((head)->tqh_last))->tqh_last)
#define TAILQ_PREV(elm, headname, field) \
    (((struct headname *) (void *) ((elm)->field.tqe_prev))->tqh_last)
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head); \
        (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head) && \
        ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head); \
        (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head) && \
        ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \

```

```

    QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
        &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

```

```

#define TAILQ_INSERT_TAIL(head,
elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
    *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \

```

```

        (head)->tqh_last = &(elm2)->field.tqe_next; \
        (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
        *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type) \
struct name { \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type) \
struct { \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
    == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \

```

```

} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))

```

```

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
(STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *)((void *) \
    ((char *)((head)->stqh_last) - offsetof(struct type, field))))

#ifndef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros
 * for compatibility but because of pointer aliasing issues their use
 * is discouraged!
 */

/*
 * __launder_type(): We use this ugly hack to work around the the compiler
 * noticing that two types may not alias each other and elide tests in code.
 * We hit this in the CIRCLEQ macros when comparing 'struct name *' and
 * 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
 * 4.8) declare these comparisons as always false, causing the code to
 * not run as designed.
 *
 * This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
 *
 * If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
 * this by changing the head/tail sentinel values, but see the note above
 * this one.
 */
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

#endif

```

```

#ifdef QUEUEDEBUG

```

```

#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
    QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
        __FILE__, __LINE__); \
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
    QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_last != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
(elm)->field.cqe_next = (void *)1L; \
(elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

```



```

#define CIRCLEQ_ENTRY(type) \
struct { \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

/*
 * Circular queue functions.
 */

#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head, \
listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

```

```

} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next \
    = CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */

#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__laundry_type(head))

```

```

/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_last) \
     : \
     (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl

```

```

}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

```

```

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

```

```

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
data structure. Every operation
* on the tree causes a splay to happen. The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree. On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as  $O((m + n) \lg n)$ . The
* amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute. It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as  $O(\lg n)$ .
* The maximum height of a red-black tree is  $2 \lg (n+1)$ .

```

```

*/

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \

```

```

(head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
    \
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
    \
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
    \
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \

```



```

{
    \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{
    \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if(__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field);\
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{
    \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
        }

```

```

    SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
} \
return (elm); \
} \
return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
\
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
\
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head,
    &__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \

```

```

{
    \
    struct type __node, *__left, *__right, *__tmp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while (1) {
        \
        if (__comp < 0) {
            \
            __tmp = SPLAY_LEFT((head)->sph_root, field);\
            if (__tmp == NULL)
                break; \
            if (__comp < 0){
                \
                SPLAY_ROTATE_RIGHT(head, __tmp, field);\
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) {
            \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            if (__tmp == NULL)
                break; \
            if (__comp > 0) {
                \
                SPLAY_ROTATE_LEFT(head, __tmp, field);\
                if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

```

```

#define SPLAY_NEGINF -1

```

```

#define SPLAY_INF 1

```

```

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)

```

```

#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)

```

```

#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)

```

```

#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)

```

```

#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))

```

```

#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

```

```

#define SPLAY_FOREACH(x, name, head) \

```

```

    for ((x) = SPLAY_MIN(name, head); \

```

```

        (x) != NULL; \

```

```

(x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL \
}

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif
#endif

```

```

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \

```

```

attr void name##_RB_INSERT_COLOR(struct name *, struct
type *); \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *); \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *); \
attr struct type *name##_RB_PREV(struct type *); \
attr struct type *name##_RB_MINMAX(struct name *, int); \
\

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
struct type
*parent, *gparent, *tmp; \
while ((parent = RB_PARENT(elm, field)) != NULL && \
RB_COLOR(parent, field) == RB_RED) { \
gparent = RB_PARENT(parent, field); \
if (parent == RB_LEFT(gparent, field)) { \
tmp = RB_RIGHT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
RB_COLOR(tmp, field) = RB_BLACK; \
RB_SET_BLACKRED(parent, gparent, field);\
elm = gparent; \
continue; \
} \
if (RB_RIGHT(parent, field) == elm) { \
RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = parent; \
parent = elm; \
elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
tmp = RB_LEFT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
RB_COLOR(tmp, field) = RB_BLACK; \

```

```

RB_SET_BLACKRED(parent, gparent, field);\
elm = gparent; \
continue; \
} \
if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_LEFT(head, gparent, tmp, field);\
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\
                    RB_COLOR(tmp, field) = RB_RED; \
                    RB_ROTATE_RIGHT(head, tmp, oleft, field);\

```

```

    tmp = RB_RIGHT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_RIGHT(tmp, field)) \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_LEFT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field); \
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\
            tmp = RB_LEFT(parent, field); \
        } \
        RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
        RB_COLOR(parent, field) = RB_BLACK; \
        if (RB_LEFT(tmp, field)) \
            RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        elm = RB_ROOT(head); \
        break; \
    } \
} \
} \
if

```



```

(elm)    \
    RB_COLOR(elm, field) = RB_BLACK; \
}      \
    \
attr struct type *    \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{      \
    struct type *child, *parent, *old = elm; \
    int color;    \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else {      \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
= child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\
        else \
            RB_RIGHT(RB_PARENT(old, field), field) = elm;\
        RB_AUGMENT(RB_PARENT(old, field)); \
    } else \
        RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) { \
        left = parent; \

```

```

do { \
    RB_AUGMENT(left); \
} while ((left = RB_PARENT(left, field)) != NULL); \
} \
goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \

```

```

    RB_RIGHT(parent, field) = elm; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head,
elm); \
return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp = \
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
} \

```

```

    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
    if (RB_RIGHT(elm, field)) {    \
        elm = RB_RIGHT(elm, field);    \
        while (RB_LEFT(elm, field))    \
            elm = RB_LEFT(elm, field);    \
    } else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
            elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_PREV(struct type *elm)    \
{    \
    if (RB_LEFT(elm, field)) {    \
        elm = RB_LEFT(elm, field);    \
        while (RB_RIGHT(elm, field))    \
            elm = RB_RIGHT(elm, field);    \
    }    \
    else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
            elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
attr struct type *    \
name##_RB_MINMAX(struct name *head, int val)    \

```

```

{
    \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) {
        \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
    for ((x) = RB_MIN(name, head); \
        (x) != NULL; \
        (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
    for ((x) = (y); \
        ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
        (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
    for ((x) = RB_MIN(name, head); \
        ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
        (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
    for ((x) = RB_MAX(name, head); \
        (x) != NULL; \
        (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
    for ((x) = (y); \

```

```

((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

```

1.24 ca-certificates 20250911-r0

1.24.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/HARICA_TLS_RSA_Root_CA_2021.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/QuoVadis_Root_CA_2.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TWCA_Global_Root_CA.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TrustAsia_TLS_RSA_Root_CA.crt-----BEGIN CERTIFICATE-----

```

Found in path(s):

*

No license file was found, but licenses were detected in source scan.

```

-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBAUA
MEYx CzAJBgNVBAYTAkJFMRkwFwYDVQQKEExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTE5MDMy

```

MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwggLiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNl5brtQyYdpokNRbopiLKkHWPd08EsCvEJ
OaFV6Wc0dwxu5FUdUiXSE2te4R2pt32Jm18Nnp8semNgQB+msLZ4j5lUlgHYruQG
vGIFAha/r6gjA7aUD7xubMLL1aa7DOn2wQL7ld5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHlf8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+CtI4wSE
y132tGqzZfxCnlEmIyDLPRt5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412IPFzYE
+cCQYDdF3uYM2HSNrpYibXRdQr4G9dlkgbIQRImwTDsHTUB+JMWKmlJ5jqSngiCN
l/onccnfxkF0oE32kRbcRoxfKWMxWxEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvqqo6GDoLclcqUC
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLltbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFxzHmuEajwfmf3lH7wvqMxX63bElaZHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczzF2sUyQQCPtlk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQrQ6lbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQRzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvweyqiv5
u+YfjyW6hY0XHgL+XVAEV8/+LbzvXMAaq7afJMbfc2hIkCwU9D9SGuTSyxTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzrnE6JwYZ+vj+vYxXX4M2bUdGc6
N3ec592kd3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgxT7PntgMTzlSdriVZzH81Xwj3QEUxeCp6
-----END
CERTIFICATE-----

Found in path(s):

*

/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/GlobalSign_Root_R46.crt

1.25 libssl3 3.5.4-r0

1.25.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy,

distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use
- in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.
- d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the

Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 1, or (at your option)  
any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free  
Software
```

```
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of

the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate

manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.26 musl 1.2.5-r10

1.26.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * iconv.c
 * Implementation of SUSv4 XCU iconv utility
 * Copyright © 2011 Rich Felker
 * Licensed under the terms of the GNU General Public License, v2 or later
 */
```

Found in path(s):

* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/iconv.c

No license file was found, but licenses were detected in source scan.

```
/*_
 * Copyright (c) 1996, 1998 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by J.T. Conklin.
 *
 * Mostly rewritten to be used in Alpine Linux (with musl c-library)
 * by Timo Teräs.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 * NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
```

Found in path(s):

* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getconf.c

No license file was found, but licenses were detected in source scan.

/*_

* Copyright (c) 2004-2006 The NetBSD Foundation, Inc.

* All rights reserved.

*

* This code is derived from software contributed to The NetBSD Foundation

* by Luke Mewburn.

* Timo Teräs cleaned up the code for use in Alpine Linux with musl libc.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

*

* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS

* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED

*

TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS

* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

* POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getent.c

1.27 aws-sdk-go-v2-internal-ini 1.8.4

1.27.1 Available under license :

GetLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

```

```

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```



```

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}

```

```

}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
}

```

```

if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }
}

```

```

}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

```



```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}

```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

```

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //

```

```

// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,

```

```

options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {

```

```

    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {

```

    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
    "type": "structure",
    "members": {
      "DirectoryId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
        "traits": {
          "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {

```

```

    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  },
  "DomainIpv4List": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainIpv6List": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {

```

```

"smithy.api#enum": [
  {
    "name": "SELF_MANAGED",
    "value": "SELF_MANAGED",
    "documentation": "SELF_MANAGED type of Active Directory"
  },
  {
    "name": "AWS_MANAGED",
    "value": "AWS_MANAGED",
    "documentation": "AWS_MANAGED type of Active Directory"
  }
]
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        }
    },
}

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  }

```

```

},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":

```



```

{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>"
            }
        }
    }
}

```

resource that was deleted.</p>"

```
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",

```

```

        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {

```



```

    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {

```

```

    "type": "list",
    "member": {

"target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
},

```

```

{
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
        ]
    },
    "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager User Subscriptions",
    "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
            "Region": {
                "builtIn": "AWS::Region",
                "required": false,

```



```

        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    }
                ]
            }
        ]
    }
]

```

```

        },
        true
    ]
}
],
"error": "Invalid Configuration: FIPS and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ],
    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ],
        }
      ],
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ],
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsFIPS"
        ]
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}

```

```

    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

```

```

"expect": {
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    }
},
"params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
}
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

```



```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",

```

```

        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
  },

```

```

        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                },
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                },
            },
            "traits": {
                "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
            },
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
            "type": "structure",
            "members": {
                "IdentityProviderArn": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                    },
                },
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                },
            },
        },
    },

```

```

    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
      }
    ]
  }
}

```



```

        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,

```

```

        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request

```

```

criteria.</p>",
    "smithy.api#required": {}
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
},

```

```

"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

```

```

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ]
  }
}

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a product for this user.</p>",
      "smithy.api#required": {}
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
        "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        }
    ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      }
    }
  }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",

```

```

"members": {
  "Subnets": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
      "smithy.api#length": {
        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {

```



```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
  },

```

```

"value": {
  "target": "smithy.api#String"
},
"traits": {
  "smithy.api#length": {
    "min": 0,
    "max": 50
  },
  "smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    }
  },

```

```

    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "TagKeys": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related

```

configuration settings. You can update any combination of settings in a single operation such as the:

- Subnets which you want to add to provision VPC endpoints.
- Subnets which you want to remove the VPC endpoints from.
- Security group ID which permits traffic to the VPC endpoints.

```

    "smithy.api#required": {}
  }

  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)

```



```

if err != nil {
    return nil, err
}

out := result.(*DisassociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}

```



```

}
}
name: License Scan

on: [pull_request]

permissions:
  contents: read

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit

```

```

run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
  are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

//

```

```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```



```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more

```

```

// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput

```

```

struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
}

```

```

if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack,
options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
}

```



```

    if err = addSpanBuildRequestEnd(stack); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.

```

```

//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {

```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseConsumptionToken *string
```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```



```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

```

```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    }
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],

```

```

    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
          "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
      }
    },
  },

```

```

    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      }
    }
  }
}

```



```

"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*"
    ],

```

```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    }
},
"rules": [

```

```

{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Endpoint"
        }
      ]
    }
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{

```

```

        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                {

"ref": "UseFIPS"
                },
                true
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],

    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"

                                },

                                "supportsFIPS"
                            ]
                        }
                    ]
                }
            ],
        },
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}
]
}
],
"rules": [
    {
        "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {

"conditions": [

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {

"fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    }
},
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

"properties": {},

                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    }
  ]
}

```

```

    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
      }
    }
  },
  {
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    }
  },
  {
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}

```



```

    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
},
{
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
}

```

```

    }
  },

  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
}

```

```

        }
    ],
    "version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto

```

```

streamline results:
<ul>
  <li>
    <p>
      <code>AccountID</code>
    </p>
  </li>
  <li>
    <p>
      <code>AmiID</code>
    </p>
  </li>
  <li>
    <p>
      <code>DualSubscription</code>
    </p>
  </li>
  <li>
    <p>
      <code>InstanceID</code>
    </p>
  </li>
  <li>
    <p>
      <code>InstanceType</code>
    </p>
  </li>
  <li>
    <p>
      <code>ProductCode</code>
    </p>
  </li>
  <li>
    <p>
      <code>Region</code>
    </p>
  </li>
  <li>
    <p>
      <code>Status</code>
    </p>
  </li>
  <li>
    <p>
      <code>UsageOperation</code>
    </p>
  </li>
</ul>
<p>For each filter, you can use one of the following <code>Operator</code> values to define the behavior of the filter:
<ul>
  <li>
    <p>
      <code>contains</code>
    </p>
  </li>
  <li>
    <p>
      <code>equals</code>
    </p>
  </li>
  <li>
    <p>
      <code>Notequal</code>
    </p>
  </li>
</ul>
}
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit. Increase this limit if parameters in request increases."
},
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
            "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {

```

```

    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n
<code>contains</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>equals</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>NotEqual</code>\n        </p>\n        </li>\n    </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            }
        }
    }
}

```



```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {

```

  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],

```

```

    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  }

```



```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",

```

```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring

```



```

    }
  },
  "SubscriptionProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {

```



```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    }
  }
}

```

```

    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {

```

"smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

```

    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  }
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
  {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
    "type": "structure",

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

```



```

        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
            {}
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",

```



```

        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }

    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.28 protobuf-go 1.36.10

1.28.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.29 iputils-clockdiff 20240905-r0

1.29.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any

patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any

part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium

customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions

of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.30 iputils-arping 20240905-r0

1.30.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices

stating that you changed the files and the date of any change.

b) You must cause

any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing

or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License

may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute  
it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may

be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.31 apk-tools 2.14.9-r3

1.31.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the

freedom

to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for

each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms

of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to

copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims;

this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when

it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.32 kelseyhightower-envconfig 1.4.0

1.32.1 Available under license :

Copyright (c) 2013 Kelsey Hightower

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.33 go.yaml.in/yaml/v2 2.4.3

1.33.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the

Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

- (a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.34 openssl 3.5.4-r0

1.34.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether

gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

- a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and
- b) cause the whole of any work that you distribute or publish, that

in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard

libraries that
accompany the operating system on which the executable
file runs, or for standard header files or definitions files that
accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original
licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution
conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free
Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the
commands you use may be called something other than `show w' and `show
c'; they could even be mouse-clicks or menu items--whatever suits your
program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here a sample; alter
the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified

Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names,
and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not
be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.35 bash 5.2.37-r0

1.35.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

=====

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work

released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed

this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an

exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to

produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.

This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this

License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must

suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;

the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation

by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept

this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a

patent against the party.

If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.

If the

Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
```

```
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/licenses/why-not-lgpl.html>>.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We

have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some

word
processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The
Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements"

or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently

reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or

in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively

when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer

to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to

this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,

@c hence no sectioning command or @node.

@display

Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document

must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for

images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `La@TeX{}` input format, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has

no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps,

when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,
you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single

copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole

aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the

licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component

(kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under

the conditions stated below. Sublicensing is not allowed;
section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the

Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization

keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own

removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent

(such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any

author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least

the "copyright" line and a pointer to where
the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to
do, use the GNU Lesser General
Public License instead of this License. But first, please read
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.
Unless otherwise stated, all files in this directory are Copyright (C)
1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,

2004,2005,2006,2007,2008,2009,2010,2011

Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

1.36 go.yaml.in-yaml 2.4.3

1.36.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.37 pam 1.7.0-r4

1.37.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkbuidl-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuidl-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildid="$_srcdir/$_pkgreal-$pkgver"
```

```
build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
    INSTALLDIRS=vendor \
    NO_PACKLIST=1 \
    NO_PERLLOCAL=1
    make
```



```

}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        --sysconfdir=/etc \
        --mandir=/usr/share/man \
        --sourcedir=/usr/share/$pkgname
    make
}

package() {
    make DESTDIR="$pkgdir" install
    install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE

```

```
}
```

```
sha512sums="
```

```
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6  
8d3390143e0e62d5e3d210ccfb61294
```

```
bsd-games-3.3.tar.gz
```

```
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0  
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
```

```
"
```

```
MIT License
```

```
Copyright (c) 2023 Leaf Corcoran
```

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
```

```
pkgname=libart-lgpl
```

```
pkgver=2.3.21
```

```
pkgrel=8
```

```
pkgdesc="A library for high-performance 2D graphics"
```

```
url="https://www.levien.com/libart/"
```

```
arch="all"
```

```
options="!check" # No test suite.
```

```
license="LGPL-2.0-or-later"
```

```
subpackages="$pkgname-dev"
```

```
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
```

```
"
```

```
builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {
```

```
update_config_sub
```

```
default_prepare
}
```

```
build() {
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr
make
}
```

```
package() {
make DESTDIR="$pkgdir" install
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

```
# Contributor: Patrycja Rosa <alpine@ptrcnll.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
```

```

pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
    mkdir -p "$pkgdir"
}

```

```
_subpkg() {
local type=${subpkgname/$pkgname-/}
pkgdesc="$pkgdesc ($type)"
install_if="$pkgname=$pkgver-r$pkgrel"
mkdir -p "$subpkgdir"/usr/share/spdx
cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}
```

```
list() {
pkgdesc="$pkgdesc (licence list)"
mkdir -p "$subpkgdir"/usr/share/spdx
local i; for i in "$builddir"/text/*.txt;
do
local license=${i##*/}
echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
done
}
```

```
sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
The text rotation code in w_rottext.c was written by Alan Richardson.
The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-

bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

Contributor: Dhruvin Gandhi <contact@dhruvin>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

_pkgname=license_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools_scm py3-gp517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/_pkgname/_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\$_pkgname-\$pkgver"

```
build() {
    gp517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}
```

```
check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}
```

```
package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}
```

```
sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"
```

```
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */
```

```
/*
```

```
* Copyright (c) 1991, 1993
```

```
* The Regents of the University of California. All rights reserved.
```

```

*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
* BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
*/

```

```

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

```

```

/*
* This file defines five types of data structures: singly-linked lists,
* lists, simple queues, tail queues, and circular queues.
*
* A singly-linked list is headed by a single forward pointer. The
* elements are singly linked for minimum space and
* pointer manipulation
* overhead at the expense of O(n) removal for arbitrary elements. New
* elements can be added to the list after an existing element or at the
* head of the list. Elements being removed from the head of the list
* should use the explicit macro for this purpose for optimum
* efficiency. A singly-linked list may only be traversed in the forward
* direction. Singly-linked lists are ideal for applications with large
* datasets and few or no removals or for implementing a LIFO queue.
*
* A list is headed by a single forward pointer (or an array of forward

```

- * pointers for a hash table header). The elements are doubly linked
- * so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before
- * or after an existing element or at the head of the list. A list
- * may only be traversed in the forward direction.
- *
- * A simple queue is headed by a pair of pointers, one the head of the
- * list and the
- other to the tail of the list. The elements are singly
- * linked to save space, so elements can only be removed from the
- * head of the list. New elements can be added to the list after
- * an existing element, at the head of the list, or at the end of the
- * list. A simple queue may only be traversed in the forward direction.
- *
- * A tail queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before or
- * after an existing element, at the head of the list, or at the end of
- * the list. A tail queue may be traversed in either direction.
- *
- * A circle queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements
- can be added to the list before or after
- * an existing element, at the head of the list, or at the end of the list.
- * A circle queue may be traversed in either direction, but has a more
- * complex end of list detection.
- *
- * For details on the use of these macros, see the queue(3) manual page.
- */
- /*
- * Include the definition of NULL only on NetBSD because sys/null.h
- * is not available elsewhere. This conditional makes the header
- * portable and it can simply be dropped verbatim into any system.
- * The caveat is that on other systems some other header
- * must provide NULL before the macros can be used.
- */
- #ifdef __NetBSD__
- #include <sys/null.h>
- #endif
- #if defined(QEUEDEBUG)
- # if defined(_KERNEL)
- # define QEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
- # else


```

# include <err.h>
# define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
 * Singly-linked List definitions.
 */
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
 * Singly-linked List access methods.
 */
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \
    (var) != SLIST_END(head) && \
    ((tvar) = SLIST_NEXT((var), field), 1); \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

```

```
#define SLIST_INSERT_AFTER(slistelm, elm, field) do { \
    (elm)->field.sle_next = (slistelm)->field.sle_next; \
    (slistelm)->field.sle_next \
    = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_INSERT_HEAD(head, elm, field) do { \
    (elm)->field.sle_next = (head)->slh_first; \
    (head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE_AFTER(slistelm, field) do { \
    (slistelm)->field.sle_next = \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)
```

```
/*
```

```
* List definitions.
```

```
*/
```

```
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}
```

```
#define LIST_HEAD_INITIALIZER(head) \
{ NULL }
```

```
#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
}
```

```

struct type **le_prev; /* address of previous next element */\
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
     (var) != LIST_END(head) && \
     ((tvar) = LIST_NEXT((var), field), 1); \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
LIST_INIT((head2)); \
if (!LIST_EMPTY((head1))) \
{ \
(head2)->lh_first = (head1)->lh_first; \
LIST_INIT((head1)); \
} \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field) \
if ((head)->lh_first && \
     (head)->lh_first->field.le_prev != &(head)->lh_first) \
QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
    __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
if ((elm)->field.le_next && \
     (elm)->field.le_next->field.le_prev != \
     &(elm)->field.le_next) \
QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
    __FILE__, __LINE__); \
if (*(elm)->field.le_prev != (elm)) \
QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \

```

```

    __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
    (elm)->field.le_next = (void *)1L; \
    (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
    field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
        = (head)->lh_first) != LIST_END(head)) \
        (head)->lh_first->field.le_prev = &(elm)->field.le_next; \
    (head)->lh_first = (elm); \
    (elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
} while (/*CONSTCOND*/0)

```

```

QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
(elm2)->field.le_next->field.le_prev = \
    &(elm2)->field.le_next; \
(elm2)->field.le_prev = (elm)->field.le_prev; \
*(elm2)->field.le_prev = (elm2); \
QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head); \
    (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head) && \
    ((next = ((var)->field.sqe_next)), 1); \
    (var) = (next))

```

```

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last \
    = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if \
        (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
         == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
    if ((head)->sqh_first == (elm)) { \
        SIMPLEQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->sqh_first; \
        while (curelm->field.sqe_next != (elm)) \
            curelm = curelm->field.sqe_next; \
        if ((curelm->field.sqe_next = \

```

```

    curelm->field.sqe_next->field.sqe_next) == NULL) \
    (head)->sqh_last = &(curelm)->field.sqe_next; \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
    if (!SIMPLEQ_EMPTY((head2))) { \
        *(head1)->sqh_last = (head2)->sqh_first; \
        (head1)->sqh_last = (head2)->sqh_last; \
        SIMPLEQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
    (SIMPLEQ_EMPTY((head)) ? \
    NULL : \
    ((struct type \
    *) (void *) \
    ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
struct name { \
    qual type *tqh_first; /* first element */ \
    qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
    { TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *)((head)->tqh_last))->tqh_last)

```

```

#define TAILQ_PREV(elm, headname, field) \
    (((struct headname \
    *)((void *)((elm)->field.tqe_prev)))->tqh_last))
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head); \
        (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head) && \
        ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head); \
        (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head) && \
        ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
        &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \

```



```

    __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head, \
    elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \

```

```

    &(elm)->field.tqe_next; \
else \
    (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBDEBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBDEBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next) != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
    *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
    QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
    (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
    *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), \
    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
}

```

```

} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type)  \
struct name {                    \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head)  \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)  \
struct {                    \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
== STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

```

```
#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))
```

```
#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))
```

```
#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_LAST(head, type, field) \
    (STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
        ((head)->stqh_last->field.stqe_next)))
```

```
((char *)((head)->stqh_last) - offsetof(struct type, field))))
```

```
#ifndef _KERNEL
```

```
/*
```

```
* Circular queue definitions. Do not use. We still keep the macros
```

```
* for compatibility but because of pointer aliasing issues their use
```

```
* is discouraged!
```

```
*/
```

```
/*
```

```
* __launder_type(): We use this ugly hack to work around the the compiler
```

```
* noticing that two types may not alias each other and elide tests in code.
```

```
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
```

```
* 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
```

```
* 4.8) declare these comparisons as always false, causing the code to
```

```
* not run as designed.
```

```
*
```

```
* This hack is only to be used for comparisons and thus can be fully const.
```

```
* Do not use for assignment.
```

```
*
```

```
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
```

```
* this by changing the head/tail sentinel values, but see the note above
```

```
* this one.
```

```
*/
```

```
static __inline const void * __launder_type(const void *);
```

```
static __inline const void *
```

```
__launder_type(const void *__x)
```

```
{
```

```
__asm __volatile("" : "+r" (__x));
```

```
return __x;
```

```
}
```

```
#if defined(QUEUEDEBUG)
```

```
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
```

```
if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
```

```
(head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
```

```
__FILE__, __LINE__); \
```

```
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
```

```
(head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
```

```
__FILE__, __LINE__);
```

```
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
```

```
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
```

```
if ((head)->cqh_last != (elm)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ
```

```

elm last %p %s:%d", \
    (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
    (elm)->field.cqe_next = (void *)1L; \
    (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
    first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \
struct { \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

/*
 * Circular queue functions.
 */

#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_BEFORE(head,
listelm, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next
= CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

```

```

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

```

/*

* Circular queue access methods.

*/

```
#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
```

```
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
```

/* For comparisons */

```
#define CIRCLEQ_ENDC(head) (__launter_type(head))
```

/* For assignments */

```
#define CIRCLEQ_END(head) ((void *) (head))
```

```
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
```

```
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
```

```
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))
```

```
#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm)->field.cqe_next)
```

```
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \

```



```

? ((head)->cqh_last) \
:
(elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"

```

```

depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

```

```

}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifdef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0

```

```

#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
 * This file defines data structures for different types of trees:
 * splay trees and red-black trees.
 *
 * A splay tree is a self-organizing
 * data structure. Every operation
 * on the tree causes a splay to happen. The splay moves the requested
 * node to the root of the tree and partly rebalances it.
 *
 * This has the benefit that request locality causes faster lookups as
 * the requested nodes move to the top of the tree. On the other hand,
 * every lookup causes memory writes.
 *
 * The Balance Theorem bounds the total access time for m operations
 * and n inserts on an initially empty tree as  $O((m + n)\lg n)$ . The
 * amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
 *
 * A red-black tree is a binary search tree with the node color as an
 * extra attribute. It fulfills a set of conditions:
 * - every search path from the root to a leaf consists of the
 *   same number of black nodes,
 * - each red node (except for the root) has a black parent,
 * - each leaf node is black.
 *
 * Every operation on a red-black tree is bounded as  $O(\lg n)$ .
 * The maximum height of a red-black tree is  $2\lg(n+1)$ .
 */

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

```

```

#define SPLAY_ENTRY(type)  \
struct {  \
    struct type *spe_left; /* left element */  \
    struct type *spe_right; /* right element */  \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {  \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root;  \
    (head)->sph_root = tmp;  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {  \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root;  \
    (head)->sph_root = tmp;  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {  \
    SPLAY_LEFT(tmp, field) = (head)->sph_root;  \
    tmp = (head)->sph_root;  \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {  \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root;  \
    tmp = (head)->sph_root;  \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do {  \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define

```

```

    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
} \
\
/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \

```

```

    SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
} else { \
    int __comp; \
    name##_SPLAY(head, elm); \
    __comp = (cmp)(elm, (head)->sph_root); \
    if(__comp < 0) { \
        SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
        SPLAY_RIGHT(elm, field) = (head)->sph_root; \
        SPLAY_LEFT((head)->sph_root, field) = NULL; \
    } else if (__comp > 0) { \
        SPLAY_RIGHT(elm,
field) = SPLAY_RIGHT((head)->sph_root, field);\
        SPLAY_LEFT(elm, field) = (head)->sph_root; \
        SPLAY_RIGHT((head)->sph_root, field) = NULL; \
    } else \
        return ((head)->sph_root); \
} \
(head)->sph_root = (elm); \
return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
        return (elm); \
    } \
    return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \

```

```

\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\
if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKLEFT(head, __right, field);\
} else if (__comp > 0) { \
__tmp = SPLAY_RIGHT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) > 0){ \
SPLAY_ROTATE_LEFT(head, __tmp, field);\
if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKRIGHT(head, __left, field);\
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field);\
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
struct type __node, *__left, *__right, *__tmp; \
\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while (1) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if (__comp < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\

```



```

    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
        break; \
} \
SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
    __tmp = SPLAY_RIGHT((head)->sph_root, field);\
    if (__tmp == NULL) \
        break; \
    if (__comp > 0) { \
        SPLAY_ROTATE_LEFT(head, __tmp, field);\
        if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
            break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL \
}

#define RB_INIT(root) do { \

```

```

(root)->rbh_root = NULL;  \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type)  \
struct {  \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {  \
    RB_PARENT(elm, field) = parent;  \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED;  \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {  \
    RB_COLOR(black, field) = RB_BLACK;  \
    RB_COLOR(red, field) \
    = RB_RED;  \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {  \
    (tmp) = RB_RIGHT(elm, field);  \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    }  \
    RB_AUGMENT(elm);  \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else  \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else  \

```

```

(head)->rbh_root = (tmp); \
RB_LEFT(tmp, field) = (elm); \
RB_PARENT(elm, field) = (tmp); \
RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
        type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.

```

```

* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
    name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
    { \
        \
        struct type \
        *parent, *gparent, *tmp; \
        while ((parent = RB_PARENT(elm, field)) != NULL && \
            RB_COLOR(parent, field) == RB_RED) { \
            \
            gparent = RB_PARENT(parent, field); \
            if (parent == RB_LEFT(gparent, field)) { \
                \
                tmp = RB_RIGHT(gparent, field); \
                if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                    \
                    RB_COLOR(tmp, field) = RB_BLACK; \
                    RB_SET_BLACKRED(parent, gparent, field); \
                    elm = gparent; \
                    continue; \
                } \
                \
                if (RB_RIGHT(parent, field) == elm) { \
                    \
                    RB_ROTATE_LEFT(head, parent, tmp, field); \
                    tmp = parent; \
                    parent = elm; \
                    elm = tmp; \
                } \
                \
                RB_SET_BLACKRED(parent, gparent, field); \
                RB_ROTATE_RIGHT(head, gparent, tmp, field); \
            } else { \
                \
                tmp = RB_LEFT(gparent, field); \
                if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                    \
                    RB_COLOR(tmp, field) = RB_BLACK; \
                    RB_SET_BLACKRED(parent, gparent, field); \
                    elm = gparent; \
                    continue; \
                } \
                \
                if (RB_LEFT(parent, field) == elm) \
            { \
                \
                RB_ROTATE_RIGHT(head, parent, tmp, field); \
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_LEFT(head, gparent, tmp, field); \
        }

```

```

    } \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
    \
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, \
field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\
                    RB_COLOR(tmp, field) = RB_RED; \
                    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
                    tmp = RB_RIGHT(parent, field); \
                } \
                RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
                RB_COLOR(parent, field) = RB_BLACK; \
                if (RB_RIGHT(tmp, field)) \
                    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                elm = RB_ROOT(head); \
                break; \
            } \
        } else { \
            tmp = RB_LEFT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \

```

```

    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp
= RB_LEFT(parent, field); \
} \
if ((RB_LEFT(tmp, field) == NULL || \
    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
    (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
    if (RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
        struct type *oright; \
        if ((oright = RB_RIGHT(tmp, field)) \
            != NULL) \
            RB_COLOR(oright, field) = RB_BLACK;\
        RB_COLOR(tmp, field) = RB_RED; \
        RB_ROTATE_LEFT(head, tmp, oright, field);\
        tmp = RB_LEFT(parent, field); \
    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field)) \
        RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \

```

```

struct type *left;  \
elm = RB_RIGHT(elm, field);  \
while ((left = RB_LEFT(elm, field)) != NULL) \
    elm = left;  \
child = RB_RIGHT(elm, field);  \
parent = RB_PARENT(elm, field);  \
color = RB_COLOR(elm, field);  \
if (child)  \
    RB_PARENT(child, field) = parent; \
if (parent) {  \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else  \
        RB_RIGHT(parent, field) = child; \
    RB_AUGMENT(parent);  \
} else  \
    RB_ROOT(head)
= child;  \
if (RB_PARENT(elm, field) == old) \
    parent = elm;  \
(elm)->field = (old)->field;  \
if (RB_PARENT(old, field)) {  \
    if (RB_LEFT(RB_PARENT(old, field), field) == old)\
        RB_LEFT(RB_PARENT(old, field), field) = elm;\
    else  \
        RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else  \
    RB_ROOT(head) = elm;  \
RB_PARENT(RB_LEFT(old, field), field) = elm; \
if (RB_RIGHT(old, field))  \
    RB_PARENT(RB_RIGHT(old, field), field) = elm; \
if (parent) {  \
    left = parent;  \
    do {  \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
}  \
goto color;  \
}  \
parent = RB_PARENT(elm, field);  \
color = RB_COLOR(elm, field);  \
if (child)  \
    RB_PARENT(child, field) = parent; \
if (parent) {  \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else  \

```

```

    RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \
            RB_RIGHT(parent, field) = elm; \
        RB_AUGMENT(parent); \
    } else \
        RB_ROOT(head) = elm; \
    name##_RB_INSERT_COLOR(head,
elm); \
    return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \

```



```

int comp;    \
while (tmp) {    \
    comp = cmp(elm, tmp);    \
    if (comp < 0)    \
        tmp = RB_LEFT(tmp, field);    \
    else if (comp > 0)    \
        tmp = RB_RIGHT(tmp, field);    \
    else    \
        return (tmp);    \
}    \
return (NULL);    \
}    \
    \
/* Finds the first node greater than or equal to the search key */\
attr struct type *    \
name##_RB_NFIND(struct name *head, struct type *elm)    \
{    \
    struct type *tmp = RB_ROOT(head);    \
    struct type *res = NULL;    \
    int comp;    \
    while (tmp) {    \
        comp = cmp(elm, tmp);    \
        if (comp < 0) {    \
            res = tmp;    \
            tmp = RB_LEFT(tmp, field);    \
        }    \
        else if (comp > 0)    \
            tmp =    \
RB_RIGHT(tmp, field);    \
        else    \
            return (tmp);    \
    }    \
    return (res);    \
}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
    if (RB_RIGHT(elm, field)) {    \
        elm = RB_RIGHT(elm, field);    \
        while (RB_LEFT(elm, field))    \
            elm = RB_LEFT(elm, field);    \
    } else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \

```

```

while (RB_PARENT(elm, field) && \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field); \
elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \
        elm = RB_LEFT(elm, field); \
        while (RB_RIGHT(elm, field)) \
            elm = RB_RIGHT(elm, field); \
    } \
    else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                  (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
                elm = RB_PARENT(elm, field); \
            elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1

```

```

#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
for ((x) = RB_MIN(name, head); \
(x) != NULL; \
(x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0

```

```

pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-$_kernver.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

```

1.38 readline 8.2.13-r1

1.38.1 Available under license :

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,

@c hence no sectioning command or @node.

@display

Copyright @copyright{ } 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document

must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free

program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain @sc{ascii} without markup, Texinfo input format, LaTeX input format, @acronym{SGML} or @acronym{XML} using a publicly available @acronym{DTD}, and standard-conforming simple @acronym{HTML}, PostScript or @acronym{PDF} designed for human modification. Examples of transparent image formats include @acronym{PNG}, @acronym{XCF} and @acronym{JPG}. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, @acronym{SGML} or @acronym{XML} for which the @acronym{DTD} and/or processing tools are not generally available, and the machine-generated @acronym{HTML}, PostScript or @acronym{PDF} produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent

copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and

list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of

the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have

received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco,

California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts

being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies

of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the

source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any

patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and
``show c'` should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than ``show w'` and ``show c'`; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed

to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"

to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an

"aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users

beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years

and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for

the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for

sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a

covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the

combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short

notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.39 ca-certificates-bundle 20250911-r0

1.39.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

*

No license file was found, but licenses were detected in source scan.

-----BEGIN CERTIFICATE-----

```
MIIH0zCCBbugAwIBAgIIXsO3pkN/pOAwDQYJKoZIhvcNAQEFBQAwwQjESMBAGA1UE
AwwJQUUNDVlJBSVoxMRAwDgYDVQQLDAQsO1BQ0NWMQ0wCwYDVQQKDARBQ0NWMQsw
CQYDVQQGEwJFUzAeFw0xMTA1MDUwOTM3MzdaFw0zMDEyMzEwOTM3MzdaMEIxEjAQ
BgNVBAMMCUFUDQ1ZSQUlaMTEQMA4GA1UECwwHUETJQUUNDVjENMA5GA1UECgwEQUND
VjELMAkGA1UEBhMCRCVMwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCb
qau/YUqXry+XZpp0X9DZlv3P4uRm7x8fRzPCRKPfmt4ftVTdFXxpNRFvu8gMjmoY
HtiP2Ra8EEg2XPBjs5BaXCQ316PWYwLxufEBcoSwfDtNgM3802/J+Nq2DoLSRYWo
G2ioPej0RGy9ocLLA76MPhMAhN9KSMDjIgro6TenGEyxCCQ0jVn8ETdkXhBilyNpA
lHPrzg5XPAOBOP0KoVdDaaxXbXmQeOW1tDvYvEyNKKGno6e6Ak4l0Squ7a4DIrhr
IA8wKFSVf+DuzgpmndFALW4ir50awQUZ0m/A8p/4e7MCQvtQqR0tkw8jq8bBD5L/
```

0KIV9VMJcRz/RROE5iZe+OCiHAr8Fraocwa48GOEAqDGWuzndN9wrqODJerWx5eH
k6fGioozL2A3ED6XPm4pFdahD9GILBKfb6qkxkLrQaLjLUPTAYVtjrs78yM2x/47
4KElB0iryYl0/wiPgL/AlmXz7uxLaL2diMMxs0Dx6M/2OLuc5NF/1OVYm3z61PMO
m3WR5LpSLhl+0fXNWhn8ugb2+1KoS5kE3fj5ItQo05iifCHJPqDQsGH+tUtKSpa
cXpkatcnYGMN285J9Y0fkIkyF/hzQ7jSWpOGYdbhdQrqeWZ2iE9x6wQl1gpaepPl
uUsXQA+xtrn13k/c4LOsOxFwYIRKQ26ZIMApCqRAZQIDAQABo4ICyzCCAscwfQYI
KwYBBQUHAQEETBvMEwGCCsGAQUFBzACHkBoDHRwOi8vd3d3LmFjY3Y3Y3ZXMvZmls
ZWFKbWluL0FyY2hpdM9zL2NlcnRpZmljYWRvcy9yYWl6YWNjdjEuY3J0MB8GCCsG
AQUFBzABhhNodHRwOi8vb2NzcC5hY2N2LmVzMB0GA1UdDgQWBbTSh7Tj3zcnk1X2
VuqB5TbMjB4/vTAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFNKHtOPfNyeT
VfZW6oHlNsyMHj+9MIIBcwYDVR0gBiIBajCCAwwYwggFiBgRVHSAAMIIBWDCCASIG
CCsGAQUFBwICMIIBFB6CARAAQQB1AHQAAbwByAGkAZABhAGQAIABkAGUAIABDAGUA
cgB0AGkAZgBpAGMAYQBjAGkA8wBuACAAUgBhAO0AegAgAGQAZQAgAGwAYQAgAEEA
QwBDAFYAIAAoAEEAZwBlAG4AYwBpAGEAIABkAGUAIABUAGUAYwBuAG8AbABvAGcA
7QBhACAAeQAgaEMAZQByAHQAaQBmAGkAYwBhAGMAaQDzAG4AIABFAGwAZQBjAHQA
cgDzAG4AaQBjAGEALAAgaEMASQBGACAAUQA0ADYAMAAxADEANQA2AEUAKQAuACAA
QwBQAFMAIABlAG4AIABoAHQAAdABwADoALwAvAHcAdwB3AC4AYQBjAGMAAgAuAGUA
czAwBggrBgEFBQcCARYkaHR0cDovL3d3dy5hY2N2LmVzL2xlZ2lzbGFjaW9uX2Mu
aHRtMFUGA1UdHwROMEwwSqBIOEaGRGh0dHA6Ly93d3cuYWVjY3Y3Y3ZXMvZmls
aW4vQXJjaGl2b3MvY2Y2Y2dGlmawWnhZG9zL3JhaXphY2N2MV9kZXIuY3JsMA4GA1Ud
DwEB/wQEAwIBBjAXBgNVHREEDAOgQxhY2N2QGFjY3Y3Y3ZXMwDQYJKoZIhvcNAQEF
BQADggIBAxcAp/n/UNnSEQU5CmH7UwoZtCPNdpNYbdKI02125DgBS40xnnQ8pdp
D70ER9m+27Up2pvZrqmZ1dM8MJP1jaGo/AaNRPTKFPv8M9xii6g3+CfYCS0b78gU
JyCpZET/LtZ1qmxNYEAZSUNUY9rizLpm5U9EelvZaoErQNV/+QEnWCzI7UiRfD+m
AM/EKXMRNt6GGT6d7hmKG9Ww7Y49nCrADdg9ZuM8Db3VlFzi4qc1GwQA9j9ajepD
vV+JHanBsMyZ4k0ACtrJJ1vnE5Bc5PUzolVt3OAJTS+xJlsndQAjxGJ3KQhfnlms
tn6tn1QwIgpBHnFk/vk4CpYY3QIUrCPLBhwepH2NDd4nQeit2hW3sCPdK6jT2iWH
7ehVRE2I9DZ+hJp4rPcOVkkO1jMl1oRQQmwgEh0q1b688nCBpHBvgvW1m54ERL5h
I6zppSSMEYCUWqKiuUnSwdzRp+0xESyeGabu4VXhwOrPDYTkF7eifKXeVSUG7szA
h1xA2syVP1XgNce4hL60Xc16gwFy7ofmXx2utYXGJt/mwZrpHgJHnyqobalbz+xF
d3+YJ5oyXSrjhO7FmGYvliAd3djDJ9ew+f7Zfc3Qn48LFFhRny+Lwzgt3uiP1o2H
pPVWQxaZLPSkVrQ0uGE3ycJYgBugl6H8WY3pEfbRD0tVNEYqi4Y7

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgzCCA2ugAwIBAgIPXZONMGc2yAYdGsdUhGkHMA0GCSqGSIb3DQEBCwUAMDsx
CzAJBgNVBAYTAkVtMREwDwYDVQQKDAhGtK1ULVJDTTEZMBcGA1UECwwQQUMgUkFJ
WiBGTK1ULVJDTTAAeFw0wODEwMjIxNTU5NTZaFw0zMDAxMDEwMDAwMDBaMDsxCzAJ
BgNVBAYTAkVtMREwDwYDVQQKDAhGtK1ULVJDTTEZMBcGA1UECwwQQUMgUkFJWiBG
TK1ULVJDTTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALpxgHpMhm5/
yBNtwMZ9HACXjyWMI7sQmkCpGreHiPibVmr75nuOi5KOPYvdWRHbNi63URcfqQgf
BBckWko3Shjf5TnUV/3XwSyRAZHiItQDwFj8d0fsjz50Q7qsNI1NOHZnjdIbzAz
WHFctPvrbtQBULgTfmXko0nRIBnuvMApGGWn3v7v3QqQIecaZ5JCEJhfTzC8PhxF
tBDXaEAUwED653cXeuYLj2VbPNmaUtu1vZ5Gzz3rkQUcWJaydkxNEJY7kvqcw+Z
374jNUUeAlz+taibmSXXaXvMiwzn15Cou08YfxGyqxRxqAQVKL9LFwag0Jl1mpdIC
IfkYtwb1TplvqKtMUejPUBjFd8g5CSxJkJKZqLsXF3mwWsXmo8RZZUc1g16p6DUL
mbvkzSDGm0oGOBVo/CK67lWMK07q87Hj/LaZmtVC+nFNCM+HHmpxffenTtOmlcYF7

cqQZAZ2cC4Ffc0m6p6zzBE57lgvsEeBbphzOG9INgxwruJ4dfkUyYA8H6XdYfp9q
yGFOtibBTI3/TO80sh9l2Ll49a2pcbnvT1gdpd50IJeh7WhM3pIXS7yr/2WanvtH
2Vdy8wmhrnZEE26cLUQ5vPnHO6RYPUG9tMJJo8gN0pcvB2VSAKduyK9o7PQUlrZX
H1bDOZ8rbeTzPvY1ZNoMHKGEsy9LS+IsJJ1tk0DrtSOOMspvRdOoiXsezx76W0OL
zc2oD2rKDF65nkeP8Nm2CgtYZRczuSPkdxl9y0oukntPLxB3sY0vaJxizOBQ+OyR
p1RMVwnVdmPF6GUE7m1qzwmd+nxPrWAI/VaZDxUse6mAq4xhj0oHdkLePfTdsiQz
W7i1o0TJrH93PB0j7IKppuLIBkwC/qxcmZkLLxCKpvR/1Yd0DVlJRfbwcVw5Kda/
SiOL9V8BY9KHcyi1Swr1+KuCLH5zJTIdC2MKF4EA/7Z2Xue0sUDKlBvVgFHlSFJn
LNJhiQcND85Cd8BEc5xEUKDbEAotlRyBr+Qc5RQe8TZBAQIvfXOn3kLMTOMJDVb3
n5HUA8ZsyY/b2BzgQJhdZpmYgG4t/wHFzstGH6wCkPmrqKEPMVOHj1tyRRM4y5B
u8o5vzY8KhmQYdOpc5LMnndkEl/AgMBAAGjYzBhMB8GA1UdIwQYMBaAFJxf0Gxj
o1+TypOYCK2Mh6UsXME3MB0GA1UdDgQWBBSX9BsY6Nfk8qTmAitjlelLFzBNzAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOC
AgEATH65isagmD9uw2nAalxJUqzLK114OMHVVISfk/CHGT0sZonrDUL8zPB1hT+L
9IBdeeUXZ701guLyPI59WzbLWoAAKfLOKyzxj6ptBZNscsdW699QIyjlIRRA96Gej
rw5VD5AJYU9LWaL2U/HANeQvwSS9eS9OICI7/RogsKQOLHDD+4E5UGUcjohybK
pFtqFiGS3XNgnhAY3jyB6ugYw3yJ8otQPr0R4hUDqDZ9MwFsSBXXiJCZBMXM5gf0
vPSQ7RPI6ovDj6MzD8EpTBNO2hVWcXNygID2mjN8orGoGjR0ZVzO0eurU+AagNjq
OknkJjCb5RyKqKkVMoaZkgoQI1YS4PbOTOK7vtuNknMBZi9iPrJyJ0U27U1W45eZ
/zo1PqVUSIJZS2Db7v54EX9K3BR5YLZrZAPbFYPhor72I5dQ8AkzNqdxliXzuUJ9
2zg/LFis6ELhDjtTO0wugumDLmsx2d1Hhk9tl5EuT+IocTUW0fJz/iUrB0ckYyfl
+PbZa/wSMVYIwFNCr5zQM378BvAxRAMU8Vjq8moNqRGyg77FG8H6lnc04g175x2
MjxNBiLOFeXdntiP2t7SxDnlF4HPOEfrf4htWRvfn0IUrn7PqLBmZdo3r5+qPeoo
tt7VMVgWglvquxl1AnMaykgaIZOQCo6ThKd9OyMYkomgjaw=
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFuzCCA6OgAwIBAgIIVwoRI0LE48wwDQYJKoZIhvcNAQELBQAwazELMAkGA1UE
BhMCSVQxDjAMBgNVBACMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8w
MzM1ODUyMDk2NzEnMCUGA1UEAwweQWN0YXxpcyBBdXR0ZW50aWNhdGlvbiBSb290
IENBMB4XDTEyMDk2NzEnMCUGA1UEAwweQWN0YXxpcyBBdXR0ZW50aWNhdGlvbiBSb290IENB
SVQxDjAMBgNVBACMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8wMzM1
ODUyMDk2NzEnMCUGA1UEAwweQWN0YXxpcyBBdXR0ZW50aWNhdGlvbiBSb290IENB
MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICCGKCAgEAp8bEpSmkLO/lGMWwUKNv
UTufClrJwkg4CsIcoBh/kbWHuUA/3R1oHwiD1S0eiKD4j1aPbZkCkpAW1V8IbInX
4ay8IMKx4INRimlNAJZaby/ARH6jDuSRzVju3PvHHkVH3Se5CAGfpiEd9UEtL0z9
KK3giq0itFZljoZUj5NDKd45RnijMCO6zfB9E1fAXdKDa0hMxKufgFpbOr3JpyI/
gCczWw63igxdBzclY2zSekciRDXFzMwuJt0q7bd9Zg1fYVEiVRvjRuPjPdA1Yprb
rxTIW6HMiRvhMCb8oJsfgadHHwTrozmSBp+Z07/T6k9QnBn+locePGX2oxgk4YQ
51Q+qDp2JE+BicXjDwL4k5RHILv+1A7TaLndxHqEguNTVHnd25zS8gebLra8Pu2F
be8IEfKXGkJh90qX6IuxEAF6ZYGYojn9zz/GPvG8VqLWeICrHuS0E4UT11F9gxe
KF+w6D9Fz8+vm2/7hNN3WpVvrJSEnu68wEqPSP4RCHiMUVhUE4Q2OM1fEwZtN4F
v6MGn8i1zeQf1xcGDXqVdFUNaBr8EBtiZJ1t4JWgw5QHVw0U5r0F+7if5t+L4sbn
fPb2U8WANFAoWPASUHEXMLrmeGO89LKtmyuy/uE5jF66CyCU3nuDuP/jVo23Eek7
jPKxwV2dpAtMK9myGPW1n0sCAwEAAANjMGEwHQYDVR0OBBYEFfLYiDrIn3hm7Ynz
ezhwLMkCAjBQMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUUtiOsifeGbt
ifN7OHCuYQICNtAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4ICAQAL

e3KHwGCmSUyIWOYdiPcUZEim2FgKDk8TNd81HdTtBjHlgT5q1d07GjLukD0R0i70
jsNjLiNmsGe+b7bAEzlgqqI0JZN1Ut6nna0Oh4lScWoWPBkdg/iaKWW+9D+a2fDz
WochcYBNy+A4mz+7+uAwTc+G02UQGRjRlwKxK3JCaKygvU5a2hi/a5iB0P2avl4V
SM0RFbnAKVy06Ij3Pjaut2L9HmLecHgQHEhb2rykOLpn7VU+Xlff1ANATIGk0k9j
pwlCCRT8AKnCgHNPLsBA2RF7SOp6AsDT6ygBJlh0wcBzIm2Tlf05fbsq4/aC4yyX
X04fkZT6/iyj2HYauE2yOE+b+h1IYHkm4vP9qdCa6HCPSXrW5b0KDtst842/6+Ok
fcvHlXHo2qN8xcL4dJIEG4aspCJTQLas/kx2z/uUMsA1n3Y/buWQbqCmJqK4LL7R
K4X9p2jlugErsWx0Hbhzelefut8cl8ABMALJ+tgulHPPAUJ4lueAI3jZm/zelo0btU
ZCzJJ7VLkn5l/9Mt4blOvH+kQSGQQXemOR/qnuOf0GZvBeyqdn6/axag67XH/JJU
LysRjYU3eExRarDzzFhdFPFqSBX/wge2sY0PjlxQRrM9vwGYT7JZVEc+NHt4bVaT
LnPqZih4zR0Uv6CPLY64Lo7yFIrM6bV8+2ydDKXhlg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDTDCCAjSgAwIBAgIIId3cGJyapsXwwDQYJKoZIhvcNAQELBQAwwRDELMakGA1UE
BhMCVVMxjFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDDBZBZmZpcmlUcnVz
dCBDdb21tZXJjaWFsMB4XDTEwMDEyYWNloXDTMwMTIzMTE0MDYwNlowRDEL
MAkGA1UEBhMCVVMxjFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDDBZBZmZp
cm1UcnVzdCBDdb21tZXJjaWFsMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEA9htPZwcroRX1BiLLHwGy43NFBkRJLLtJJRTWzsO3qyxPxEylFf6EqdbDuKP
Hx6GGaeqtS25Xw2Kwq+FNXkyLbscYjfySvtKPcrNcV/pQr6U6Mje+SJIZMblq8Yr
ba0F8PrVC8+a5fBQpIs7R6UjW3p6+DM/uO+Zl+MgwdYoic+U+7lF7eNAFxFHudPAL
MeIrJmqbTFeurCA+ukV6Bf09m2kVrn1OIGPENXY6BwLJN/3HR+7o8XYdcxXyl6S1
yHp52UKqK39c/s4mT6NmgTWvRLpUHhwwMmWd5jyTXlBOeuM61G7MGvv50jeuJCqr
VwMiKA1JdX+3KNp1v47j3A55MQIDAQABo0IwQDAdBgNVHQ4EFgQUZPGU4teyq8/
nx4P5ZmVvCT2lI8wDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwDQYJ
KoZIhvcNAQELBQADggEBAFis9AQOzcAN/wr91LoWXym9e2iZWEnStB03TX8nfUYG
XUPGhi4+c7lmfU+TqbbEKpqrIZcUsd6M06uJFdhRjNTxFq7YpFzUf1GO7RgBsZNj
vbz4YYCanrHOQnDiqX0GJX0nof5v7LMeJNrjS1UaADs1tDvZ110w/YETifLCBivt
Z8SOyUOyXGsviQK8YvxO8rUzqrJv0wqiUOP2O+guRMLbZjipM1ZI8W0bM40NjD9g
N53Tym1+NH4Nn3J2ixufcv1SNUFFApYvHLKac0khsUlhRUe072o0EclNmsxZt9YC
nlpOZbWUrhvfKbAW8b8Angc6F2S1BLUjIZkKlTuXfO8=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDTDCCAjSgAwIBAgIIIfE8EORzUmS0wDQYJKoZIhvcNAQEFBQAwwRDELMakGA1UE
BhMCVVMxjFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDDBZBZmZpcmlUcnVz
dCBOZXRX3b3JraW5nMB4XDTEwMDEyYWNloXDTMwMTIzMTE0MDYwNlowRDEL
MAkGA1UEBhMCVVMxjFDASBgNVBAoMC0FmZmlybVRydXN0MR8wHQYDVQQDDDBZBZmZp
cm1UcnVzdCBOZXRX3b3JraW5nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAAtTMMxcua5Rsa2FSoUuj3mUTOWUgJnLVWREZY9nZOIG41w3SfYvm4SEHi3y
YJ0wTsyEheIszxe6/jarM3c1RNng1lho9Nuh6DtjVR6FqaYvZ/Ls6rmla1fTWcbua
kCNrmreIdIcMHI+5ni36q1Mr3Lt2PpNMCAiMHqIjHNRqrSK6mQEubWXLviRmVSRL
QESxG9fhw0XA3hA/Pe24/PHxI1Pcv2WXb9n5QHGNfb2V1M6+oF4nI979ptAmDgAp
6zxG8D1gvz9Q0twmQVGeFDdCBKNwV6gbh+0t+nvujArjqWaJGctB+d1ENmHP4ndG
yH329JKBNv3bNPFyfvMMFr20FQIDAQABo0IwQDAdBgNVHQ4EFgQUUBx/S55zawm6i

FUF0b3MgVHJ1c3RlZFJvb3QgMjAxMTENMAsGA1UECgwEQXRvczELMAkGA1UEBhMC
REUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvHTuXbyo7LjvPpvMp
Nb7PGKw+qtn4TaA+Gke5vJrf8v7MPkfoepbCJI419KkM/IL9bcFyYie96mvr54rM
VD6QUM+A1JX76LWC1BTftqlVJVfbsVD2sGBkWXppzwO3bw2+yj5vdHLqqjAqc2K+
SZFhyBH+DgMq92og3AIVDV4VavzjgsG1xZ1kCWyjWZgHJ8cblithdHFsQ/H3NYkQ
4J7sVaE3IqKHBAUsR320HLliKWYoyrfhk/WklAOZuXCfTeZI6o1Q/NnezG8HDt0L
cp2AMBYHIT8oDv3FdU9T1nSatCQujgKRz3bFmx5VdJx4IbHwLfELn8LVlhgf8FQi
eowHAgMBAAGjftB7MB0GA1UdDgQWBBSnpQaxLKYYJO7Rl+lwrrw7GWzbITAPBgNV
HRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFKelBrEspglg7tGX6XCuvDsZbNshMBgG
A1UdIAQRMA8wDQYLKwYBBAGwLQMEAQEwDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3
DQEBCwUAA4IBAQAmdzTbIEiGKkGdLD4GkGDEjKwLVlgfuXvTBznk+j57sj1O7Z8j
vZfza1zv7v1Apt+hk6EKhqzvINB5Ab149xnYJDE0BAGmuhWawyfc2E8PzBhj/5kP
DpFrdRbhIfzYJsdHt6bPWHJxfrhTZVHO8mvbaG0weyJ9rQPOLXiZNwLz6bb65pc
maHFCN795trV1lpFDMS3wrUU77QR/w4VtFX128a961qn8FYiqTxlVMYVqL2Gns2D
lmh6cYgJ4Qvh6hEbaAjMaZ7snkGeRDImeuKHCnE96+RapNLbxc3G3mB/ufNPRJLv
KrcYPqcZ2Qt9sTdBQrC6YB3y/gkRsPCHe6ed
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICFTCCA ZugAwIBAgIQPZg7pmY9kGP3fiZXOATvADAKBggqhkiOPQQDAzBMMS4w
LAYDVQQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgRUNDIFRMUyAyMDIxMQ0w
CwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAEFw0yMTA0MjIwOTI2MjNaFw00MTA0
MTcwOTI2MjJaMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBDQSBF
Q0MgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMHYwEAYHKoZI
zj0CAQYFK4EEACIDYgAEloZYKDcKZ9Cg3iQZGeHkBQcfl+3oZIK59sRxUM6KDP/X
tXa7oWyTbIOiaG6l2b4siJBzV3dscqDY4PMwL502eCdpO5KtlbgmClBk1IQ1SQ4
AjJn8ZQsb+/Xxd4u/RmAo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBRR2
KCXWfeBmmnoJsmo7jjPXNtNPojAOBgNVHQ8BAf8EBAMCAYYwCgYIKoZIzj0EAwMD
aAAwZQIwW5kp85wxtolrbNa9d+F851F+uDrNozZffPc8dz7kUK2o59JZDCaOMDtu
CCrCp1rIAjEAMEMM56PDr9NJLkaCI2ZdyQAUEv049OGYa3cpetskz2VAv9LcjBHo
9H1/IIspQuQo
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFZDCCA0ygAwIBAgIQU9XP5hmTC/srBRLYwqipDANBgqhkiG9w0BAQwFADBM
MS4wLAYDVQQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgUINBIFRMUyAyMDIx
MQ0wCwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAEFw0yMTA0MjIwOTIxMTBaFw00
MTA0MTcwOTIxMDIAMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBD
QSBSU0EgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMIICijAN
BgqhkiG9w0BAQEFAAOCAg8AMIICCGKCAgEAtoAOxHm9BYx9sKODtSjNy/BB101Z
4NH+VoyX8te9j2y3I49f1cTYQcvyAh5x5en2XssIKl4w8i1mx4QbZFc4nXUtVsYv
Ye+W/CBGvevUez8/fEc4BKkbqlLfEzfTFRVOvV98r61jx3ncCHvVoOX3W3WsgFWZ
kmGbZSoXfduP9LVq6hdKZChmFSlsAvFr1bjqM9xaZ6cF4r9lthawEO3NUDPJcFDs
GY6wx/J0W2tExn2WuZgIWwBeKQGb9Cpt0xU6kGpn8bRrZtkh68rZYnxGEFzedUln
nkL5/nWpo63/dgpnQOPF943HhZpZnmKaau1Fh5hnstVKPNe0OwANwI8f4UDErmwh
3El+fsqyjW22v5MvoVw+j8rtgI5Y4dtXz4U2OLJxpAmMkokliEjxQGMYSluMWuPD

1aUsIC+nmCjuRfzXuIgALi9C2IHVnOUTaHFFQ4ueCyE8S1wF3BqfmI7avSKecs2t
CsvMo2ebKHTEm9caPARYpoKdrCd7b/+Alun4jWq9GJAd/0kakFI3ky88Al2CdgtR
5xbHV/g4+afNmyJU72OwFW1TZQNKXkqgsqeOSQBZONXH9IBk9W6VULgRfhVwOEqw
f9DEMnDAGf/JOC0ULGb0QkTmVXYbgBVX/8Cnp6o5qtjTcNAuuuuUavpfNlbnYrX9
ivAwhZTJryQCL2/W3Wf+47BVTwSYT6RBVUKT0Gro1vP7ZeDOdcQxWQzugsgMYDNK
GbqEZycPvEJdvSRUDewdcAZfpLz6IHxV

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFdDCCA1ygAwIBAgIQVW9l47TZkGobCdFsPsBsIDANBgkqhkiG9w0BAQsFADBU
MQswCQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRI
T1JJVFkxHTAbBgNVBAMMFJEJKQ0EgR2xvYmFsIFJvb3QgQ0ExMB4XDTE5MTIxOTAz
MTYxN1oXDTQ0MTIxMjAzMTYxN1owVDELMaKGA1UEBhMCQ04xJjAkBgNVBAoMHUJF
SUpJTkcqQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDBRCSkNBIEsb2Jh
bCBsb290IENBMTCCAiiwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAPFmCL3Z
xRVhy4QEqaVpN3cdwbB7+sN3SJATcmTRuHyQNZ0YeYjJlW8R4HyDqKYDZ4/N+AZ
spDyRhySsTphzvq3Rp4Dhtczbu33RYx2N95ulpH3134rhxfVizXuhJFyV9xgw8O5
58dnJCNPYwpj9mZ9S1WnP3hkSWkSl+BMDdMJoDIwOvqfwPKcxRIqLhy1BDPpDgR
at7GGPZHOiJBhyL8xIkoVNiMpTAK+BcWyqw3/XmnkRd4OJmtWO2y3syJfQOcs4ll
5+M7sSKGjwZteAf9kRJ/sGsciQ35uMt0WwfCyPQ10WRjeulumijWML3mG90Vr4Tq
nMfK9Q7q8l0ph49pczm+LiRvRSGsxdRpJQaDrXplhRMsDQa4bHlW/KNnMoH1V6XK
V0Jp6VwkYe/iMBhORJhVb3rCk9gZtt58R4oRTklH2yiUAguUSiz5EtBP6DF+bHq/
pj+bOT0CFqMYs2esWz8sgytnOYFcuX6U1WTdno9uruh8W7TXakdl136z1C2OVnZO
z2nxbkRs1CTqjSShGL+9V/6pmTW12xB3uD1IutbB5/EjPtffhZ0nPNRAvQoMvfXn
jSXWgXSHRtQpdaJCbPdied9v3pKH9MiyRVVz99vffXQpIsHETdfg6YmV6YBW37+
WGgHqel62bno/1Afq8K0wM7o6v0PvY1NuLxxAgMBAAGjQjBAMB0GA1UdDgQWBbTF
7+3M2I0hXkjK49cULqcWk+WYATAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQE
AwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAUoKsITQfI/Ki2Pm4rzc2IInRNwPWaZ+4
YRC6ojGYWUfo0Q0lHhVBDOAqVdVXUsv45MdpoX1NcQJeXyFFYEhcCY5JEMEE3Kli
awLwQ8hOnThJdMkycFRtwUf8jrQ2ntScvd0g1lPJGKm1Vrl2i5VnZu69mP6u775u
+2D2/VnGKhs/I0qUJDAnyIm860Qkmss9vk/Ves6OF8tiwdneHg56/0OGNFK8YT88
X7vZdrRTvJez/opMEi4r89fO4aL/3Xtw+zuhTaRjAv04l5U/BXCga99igUOLtFkN
SoxUnMW7gZ/NfaXvCyUeOiDbHPwfmGcCctRzRBPbUYQaVQNW4AB+dAb/OMRyHdOo
P2gxXdMjxy6MW2Pg6Nwe0uxhHvLe5e/2mXZgLR6UcnHGCyoyx5JO1UbXHfmpGQrI
+pXObSOYqgs4rZpWDW+N8TEAiMEXnM0ZNjX+VVOg4DwzX5Ze4jLp3zO7Bkqp2IRz
znfSxqxx4VyjHQy7Ct9f4qNx2No3WqB4K/TUfet27fJhcKvLmtOJNBir+3I+17Q9
eVzYH6Eze9mCUAyTF6ps3MKCuwJXNq+YJyo5UOGwifUll35HaBC07HPKs5fRJNz2
YqAo07WjuGS3iGJCz51TzZm+ZGiPTx4SSPfsKcOYKMryMguTjClPPGAyzQWWYez
r/6zcCwupvI=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICJTCCAaugAwIBAgIQLBcIfWQqWP6FGFkGz7RK6zAKBggqhkiOPQQDAzBUMQsw
CQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRIITJJ
VFkxHTAbBgNVBAMMFJEJKQ0EgR2xvYmFsIFJvb3QgQ0EyMB4XDTE5MTIxOTAzMTgy
MVoXDTQ0MTIxMjAzMTgyMVowVDELMaKGA1UEBhMCQ04xJjAkBgNVBAoMHUJFJSUpJ

TkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDBRCskNBIEdsb2JhbCBS
b290IENBMjB2MBAGByqGSM49AgEGBSuBBAAiA2IABJ3LgJGNU2e1uVCxA/jlSR9B
IgmwUVJY1is0j8USRhTFiy8shP8sbqjV8QnjAyEUxEM9fMEsxEtqSs3ph+B99iK+
+kpRuDCK/eHeGBIK9ke35xe/J4rUQUyWPGCWwf0VHKNCMEAwHQYDVR0OBBYEFNJK
sVF/BvDRgh9Obl+rg/xI1LCRMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMAoGCCqGSM49BAMDA2gAMGUCMBq8W9f+qdJUDkpd0m2xQNz0Q9XSSpKZElAA
94M04TVOSG0ED1cxMDAtsaqdAzjbBgIxAAMvMh1PLet8gUXOQwKhbYdDFUDn9hf7B
43j4ptZLvZuHjw/l1lOWqzzIQNph91Oj9w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcjIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMiBSb290IENBMB4XDTEwMTAyNjA4MzgwM1oXDTQwMTAyNjA4MzgwM1ow
TjELMAkGA1UEBhMCTk8xHTAbBgNVBAoMFEJ1eXBhc3MgQVMtOTgzMTYzMzI3MSAw
HgYDVQQDDBdCdXlwYXNzIENsYXNzIDIGUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgoCggIBANfHXvBB9R3+0Mh9PT1aeTuMgHbo4Yf5FkNuud1g1Lr
6hxxFUi7HQfKjK6w3Jad6sNgkoaCKHOCVgb/S2TwDCo3SbXlzx87vFKu3MwZfPV
L4O2fuPn9Z6rYPnT8Z2SdIrkHJasW4DptfQxh6NR/Md+oW+OU3fUl8FVM5I+GC91
1K2GScuVr1QGbnGGE41b/+EmGVnAJLqBcXmQRFB0JJRfuLMR8SIBYaNBYYM21cHx
MIAQTn/0hpPshNOOvEu/XAFOBz3cFIqUCqTqc/sLUegTBxj6DvEr0VQVfTzh97QZ
QmdiXnfgolXsttlpF9U6r0TtSsWe5HonfOV116rLJeffawrbD02TTqigzXsu8lkB
arcNuAeBfos4GzjmCleZPe4h6KP1DBbdi+w0jpwqHAAVF41og9JwnxgIzRFo1clr
Us3ERo/ctfPYV3Me6ZQ5BL/T3jjetFPsaRyifsSP5BtwrfKi+fv3FmRmaZ9JUaLi
FRhnBkp/1Wy1TbMz4GhrXb7pmA8y1x1LPC5aAVKRCfL6o3YBkBjqhHk/sM3nhRS
P/TizPJhk9H9Z2vXUq6/aKtAQ6BXNVN48FP4YUIHZMbXb5tMOA1jrGKvNouicwoN
9SG9dKpN6nIDSdvHX1iY8f93ZHsM+71bbRuMGjeyNYmsHVee7QHJIjhdK4TWxP
AgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFMmAd+BikoL1Rpzz
uvdMw964o605MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAU18h
9bqwoII5LJKwbADJ784g7wbylp7ppHR/ehb8t/W2+xUbP6umwHJdELFx7rxP462s
A20ucS6vxOOto70MEae0/0qyexAQH6dXQbLArvQsWdZHEIjzIVEpMMpghq9Gqx3t
OluwlN5E40EIosHsHdb9T7bWR9AUC8rmyrV7d35BH16Dx7aMOZawP5aBQW9gkOLO
+fsicdl9sz1Gv7SEr5AcD48Saq/v7h56rgJKihcrdv6sVlkkLE8/trKnToyokZf7
KcZ7XC25y2a2t6hbElGftQl+Ynhw/qlqYLYdDnkM/crqJIByw5c/8nerQyIKx+u2
DISCLIBrQYoIwOula9+ZEsuK1V6ADJHgJgg2SMX6OBE1/yWDLfJ6v9r9jv6ly0Us
H8SIU653DtmadsWOLB2jutXsMq7Aqqz30XpN69QH4kj3Io6wpJ9qzo6ysmD0oyLQ
I+uUWnpp3Q+/QFesa1lQ2aOZ4W7+jQF5JyMV3pKdewlNWudLSDBaGOYKbeaP4NK7
5t98biGCwWg5TbSYWGWZizEqQXsP6JwSxeRV0mcy+rSDeJmAc61ZRpqPq5KM/p/9h
3PFaTWwyI0PurKju7koSCTxdccK+efrCh2gdC/1cacwG0Jp9VJkqyTkaGa9LKkPz
Y11aWOIv4x3kqdbQCtCev9eBCfHJxyYNrJgWVqA=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcjIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMyBSb290IENBMB4XDTEwMTAyNjA4Mjg1OFOxDTQwMTAyNjA4Mjg1OFow

hvcNAQELBQADggIBACYGXnDnZTPIgm7ZnBc6G3pmsgH2eDtpXi/q/075KMOYKmFM
tCQSin1tERT3nLXK5ryeJ45MGcipvXrA1zYObYVybqjGom32+nNjf7xueQgcnYqf
GopTpti72TVVsRHFqQOzVju5hJMiXn7B9hJSi+osZ7z+Nkz1uM/Rs0mSO9MpDpkb
lvdhuDvEK7Z4bLQjb/D907JedR+Zlais9trhxTF7+9FGs9K8Z7RiVL0J92Owk6Ka
+elSLotgEqv89WBW7xBci8QaQtyDW2QOy7W81k/BfDxujRNt+3vrMNDcTa/F1bal
TFtxyegxvug4BkihGuLq0t4SOVga/4AOgnXmt8kHbA7v/zjxmHHEt38OFdAlab0i
nSvtBfZGR6ztwPDUO+Ls7pZbkBNOHIY667DvlruWlXG68kOGdGSVyCh13x01utI3
gzhTODY7z2zp+WsO0PsE6E9312UBelYMej4hYvF/Y3EMyZ9E26gnonW+boE+18Dr
G5gPcFw0sorMwIUy6256s/daoQe/qUKS82Ail+QUoQebTnbAjn39pCXHR+3/H3Os
zM0l6W8KjptlwlCFtaOgUxLMVYdh84GuEEZhvUQhuMI9dM9+JDX6HAcOmz0iyu8x
L4ysEr3vQCj8KWefshNPZiTEUxnpHikV7+ZtsH8tZ/3zbBt1RqPlShfppNcL

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFjTCCA3WgAwIBAgIEGERM1jANBgkqhkiG9w0BAQsFADBWMQswCQYDVQQGEwJD
TjEwMC4GA1UECgwnQ2hpbmEgRmluYW5jaWFsIENlcnRpZmljYXRpb24gQXV0aG9y
aXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJPT1QwHhcNMTIwODA4MDMwNzAxWhcNMjkx
MjMxMDMwNzAxWjBWMQswCQYDVQQGEwJDTjEwMC4GA1UECgwnQ2hpbmEgRmluYW5j
aWFsIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MRUwEwYDVQQDDAxDRkNBIEVWIFJP
T1QwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDXXWvNED8fBVnVBU03
sQ7smCuOFR36k0sXgiFxEFLXUWRwFsJvAu2OFW2fvwwbwuCjZ9YMrM8irq93VCpL
TlPtUnrD7i7es3ElweldPe6hL6P3KjzJlx1qqx2hp/Hz7KDVRM8Vz3IvHWOX6Jn5
/ZOKVIBMUtRSqy5J35DNuF++P96hyk0g1CXohClTt7GIH//62pCfCqktQT+x8Rgp
7hZZLDRJGqgG16il0gNyejLi6mhNbiyWZXvKWfry4t3uMCz7zEasGPrb382KzRz
EpR/38wmnvFyXVBIWY9ps4deMm/DGIq1IY+wejfeWkU7xzbh72fROdOXW3NiGUgt
hxxG+3SYIElz8AXSG7Ggo7cbcNOIabla1jj0Ytwli3i/+Oh+uFzJlU9fpy25IGvP
a931DfScT/SyZi4QKPaXWnuWFO8BGS1sbn85WAZkgwGDg8NNkt0yxoekN+kWzqot
aK8KgWU6cMGbrU1tVMoqLUuFG7OA5nBFDWteNfB/O7ic5ARwiRilK9oKmSJgamNg
TnYGme69g60dWl0lhdLHZR4tjsbftsbbhf4oEIRUpdPA+nJCdDC7xij5aqgwJHsfV
PKPtl8MeNPo4+QgO48BdK4PRVmrJtqhUUy54Mmc9gn900PvhtgVguXDbjgv5E1hv
cWAQUhC5wUEJ73IfZzF4/5YFjQIDAQABo2MwYTAfBgNVHSMEGDAWgBTj/i39KNAL
tbq2osS/BqoFjJP7LzAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAd
BgNVHQ4EFgQU4/4t/SjQC7W6tqLEvwaqBYyT+y8wDQYJKoZIhvcNAQELBQADggIB
ACXGumvrh8vegmWPfBEp2uEcwPenStPuiB/vHiyz5ewG5zz13ku9Ui20vsXiObT
ej/tUxPQ4i9qecsAIymHjdXNYmEwnZPNdatZ8POQQaIxxffu2Bq41gt/UP+TqhdL
jOztUmCypAbqTuv0axn96/Ua4CUqmtzHQTb3yHQFhDmVOdYLO6Qn+gjYXB74BGBS
ESgoA//vU2YApUo0FmZ8/Qmkrp5nGm9BC2sGE5uPhnEFtC+NiWYzKXZUmhH4J/qy
P5Hgzg0b8zAarb8iXRvTvYUFTeGSGn+ZnzxEk8rUQElsgIfXBDrDMI1Dlb4pd19
xIsNER9Tyx6yF7Zod1rg1MvIB671Oi6ON7fQAUtDKXeMOZePglr4UeWJoBjnaH9d
Ci77o0cOPaYjesYBx4/IXr9tgFa+iiS6M+qf4TIRnvHST4D2G0CvOJ4RUHlzEhLN
5mydLlhyPDCBBPei6lmt2hkuIsKNuYyH4Ga8cyNfIWRjgEj1oDwYPZTISEEdQLpe
/v5WOaHlz16eGWRGENoXkbcFgKyLmZJ956LYBws2J+dleWCKw9cTXPhyQN9Ky8+Z
AAoACxGV2lZFA4gKn2fQ1XmxqI1AbQ3CekD6819kR5LLU7m7Wc5P/dAVUwHY3+vZ
5nbv0CO7O6l5s9UCKc2Jo5YPSjXnTkLAdc0Hz+Ys63su

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIHTCCAwwGawIBAgIQToEtioJl4AsC7j41AkblPTANBgkqhkiG9w0BAQUFADCB
gTELMaKGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEaMBGGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxJzAlBgNV
BAMTHkNPTU9ETyBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0wNjEyMDEwMDAw
MDBaFw0yOTEyMzEyMzU5NTlaMIGBMQswCQYDVQQGEwJHJQJEBMBkGA1UECBMSR3Jl
YXRlcjBNYW5jaGVzdGVyMRAwDgYDVQQHEwdTYWxmb3JkMR0wGAYDVQQKEwFDT01P
RE8gQ0EgTGltaxRlZDnMCUGA1UEAxMeQ09NT0RPIENlcnRpZmljYXRpb24gQXV0
aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0ECLi3LjkRv3
UcEbVASY06m/weaKXTuH+7uIzg3jLz8GlvCiKVCZrts7oVewdFFxze1CkU1B/qnI
2GqGd0S7WWaXUF601CwRM/aN5VCaTwwxHGzUvAhTaHYujl8HJ6jJJ3ygxaYqhZ8
Q5sVW7euNJH+1GIImGEaaP+vB+fGQV+useg2L23IwambV4EajcNxo2f8ESI33rXp
+2dtQem8Ob0y2WIC8bGoPW43nOlV4tOiJovGuFVDiOEjPqXSJDlqR6sA1KGzqSX+
DT+nHbrTUcELpNqsOO9VUCQFZUaTNE8tja3G1CEZ0o7KBWfXB3NH5YoZEr0ETc5O
nKVlrLsm9wIDAQABo4GOMIGLMB0GA1UdDgQWBBQLWOWLxkwVN6RAqTCpIb5HNlpW
/zAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQH/BAUwAwEB/zBJBgNVHR8EQjBAMD6g
PKA6hjhodHRwOi8vY3JsLmNvbW9kb2NhLmNvbS9DT01PRE9DZXJ0aWZpY2F0aW9u
QXV0aG9yaXR5LmNybDANBgkqhkiG9w0BAQUFAAOCAQEAAPpiem/Yb6dc5t3iuHXIY
SdOH5EOC6z/JqvWote9VfCFSZfnVDeFs9D6Mk3ORLgLETgdx8CPOGEIqB6BCsAv
IC9Bi5HcSEW88cbeunZrM8gALTFGTO3nnc+IIP8zwFboJIYmuNg4ON8qa90SzMc/
RxdMosIGlgnW2/4/PEZB31jiVg88O8EckzXZOFKs7sjsLjBOLDW0JB9LeGna8gl4
zJVSklBwJVmcIGfE7vmlV2H0knZ9P4SNVbfo5azV8fUZVqZa+5Acr5Pr5RzUZ5dd
BA6+C4OmF4O5MBKgXTMVBbkN+8cFduPYSO38NBejxiEovjBFMR7HeL5YYTisO+IB
ZQ==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICiTCCAg+gawIBAgIQH0evqmlAcFBUTAGem2OZKjAKBggqhkiOPQQDAzCBhTEL
MAKGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4GA1UE
BxMHU2FsZm9yZDEaMBGGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxKzApBgNVBAMT
IkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlvbiBBdXR0b3JpdHkwHhcNMDgwMzA2MDAw
MDAwWhcNMzgwMTE4MjM1OTU5WjCBhTELMAKGA1UEBhMCR0IxGzAZBgNVBAGTEkdy
ZWf0ZXIgtWTFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBGGA1UEChMRQ09N
T0RPIENBIExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlv
biBBdXR0b3JpdHkwZjAQBgcqhkiOPQIBBQUrgQQAIGNiAAQDR3svdcmCFYX7deSR
FtSrYpn1PIILBs5BAH+X4QokPB0BBO490o0JlwzgdT6+3eKKvUDYEs2ixYjFq0J
cfRK9ChQtP6IHG4/bC8vCVlbpVsLM5niwz2J+Wos77LTBumjQjBAMB0GA1UdDgQW
BBR1cacZSBm8nZ3qQUfflMRId5nTeTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQH/
BAUwAwEB/zAKBggqhkiOPQQDAwNoADBlAJEA7wNbeqy3eApyt4jf/7VGFAkK+qDm
fQjGGoe9GKhzvSbKYAydzpmfz1wPMOG+FDHqAjAU9JM8SaczepBGR7NjfrObTrdv
GDeAU/7dIOA1mjbRxxwG55tZd8/8dLDoWV9mSOdY=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF2DCCA8CgawIBAgIQTKr5yttjb+Af907YWwOGnTANBgkqhkiG9w0BAQwFADCB
hTELMaKGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4G

A1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIEExpbWl0ZWQxKzApBgNV
BAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNhdGlviBBdXR0b3JpdHkwHhcNMTAwMTE5
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBhTElMAkGA1UEBhMCR0IxGzAZBgNVBAGT
EkdyZWFOZl0ZXFwFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBgGA1UEChMR
Q09NT0RPIENBIEExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNh
dGlviBBdXR0b3JpdHkwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCR
6FSS0gpWsaWJN3Fz0RNdJkrN6N9I3AAcbxT38T6KhKPS38QVr2fcHK3YX/JSw8X
pz3jsARh7v8Rl8f0hj4K+j5c+ZPmNHRZFGvnnLOFoIJ6dq9xkNfs/Q36nGz637CC
9BR++b7Epi9Pf5l/txfnQ3K9DADWietrLNPtj5gcFKt+5eNu/Nio5Jlk2kNrYrhV
/erBvGy2i/MOjZrkm2xpmfh4SDBF1a3hDTxFYPwyllEnvGfDyi62a+pGx8cgoLEf
Zd5ICLqkTqnyg0Y3hOvozIFIQ2dOciqbXL1MGyiKXCJ7tKuY2e7gUYPCDUZObT6Z
+pUX2nzwV0E8jVHTC7ZcryxjGt9XyD+86V3Em69FmeKjWiS0uqlWPc9vqv9JWL7w
qP/0uK3pN/u6uPQLOvnoQ0IeidiEyxPx2bvhiWC4jChWrBQdnArncevPDt09qZah
SL0896+1DSJMwBGB7FY79tOi4lu3sgQiUpWak2nojkx18ZEDLXB0AuqLZxUpaVIC
u9ffUGpVrR+goyhhf3DQw6KqLcGqR84onAZFdr+CGCe01a60y1Dma/RMhnEw6abf
Fobg2P9A3fvQqoh/ozM6LlweQRGBY84YcWsr7KaKtzFcOmpH4MN5WdYgGq/yapiq
crxXStJLnsQ/LBMQeXtHT1eKJ2czL+zUdqnR+WEUwIDAQABo0IwQDAdBgNVHQ4E
FgQUu69+Aj36pvE8h16t7jiY7NkyMtQwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB
/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAARx1UaEt65Ru2yyTUEUAJNMnMvl
wFTPoCWOAvn9sKIN9SCYPBMtrFaisNZ+EZLpLrqeLppysb0ZRGxhNaKatBYSaVqM
4dc+pBroLwP0rmEdEBsqpl6xf4FpuHA1sj+nq6PK7o9mfjYcwlYRm6mnPTXJ9OV
2jeDchzTc+Cir5kDOF3VSXkAKRzH7JsgHACKaVd4sjn8OoSgtZx8jb8uk2Intzna
FxiuvTwJaP+EmzzV1gsD41eeFPfR60/IvYcjt7ZJQ3mFXLrrkguhxuhqEwWsRqZ
CuhTLJK7oQkYdQxlqHvLI7cawiiFwxv/0Cti76R7CZGYZ4wUAc1oBmpjIXUDGIk
boHGhfKppC3n9KUkEEeDys30jXIYsQab5xoq2Z0B15R97QNKyvDb6KkBPvVWmcke
jkk9u+UJueBPSZI9FoJAzMxZxuY67RluaTxslbH9qh17f4a+Hg4yRvv7E491f0yL
S0Zj/gA0QHDBw7mh3aZw4gSzQbzpgJHqZJx64SIDqZxubw5IT2yHh17zbqD5daWb
QOhTsiedSrnAdyGN/4fy3ryM7xfft0kL0fJuMAsaDk527RH89elWsn2/x20Kk4yl
0MC2Hb46TpSi125sC8KKfPog88Tk5c0NqMuRkrF8hey1FGlmDoLnzc7ILaZRfyHB
NVOFBkpdn627G190

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIB9zCCAX2gAwIBAgIQBiUzsUcdMydc+Y2aub/M+DAKBggqhkJOPQQDAzA9MQsw
CQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MRowGAYDVQQDExFDZSJ0YWlu
bHkgUm9vdCBFMTAeFw0yMTA0MDEwMDAwMDBaFw00NjA0MDEwMDAwMDBaMD0xCzAJ
BgNVBAYTAiVMTMRIwEAYDVQQKEwldZXJ0YXludHkwGjA1BgNVBAMTEUNlcnRhaW5s
eSBSb290IEUxMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAE3m/4fxzf7flHh4axpMCK
+IKXgOqPyEpeKn2IaKcBYhSRJHpcnqMXfyqGITQYUBsQ3tA3SybHGwCA6TS9YBk2
QNYphwk8kXr2vBMj3VIOBF7PyAlcGFpBMDjaIOIejeR2o0IwQDAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU8ygYy2R17ikq6+2uI1g4
hevIlgcwCgYIKoZIzj0EAwMDaAAwZQIxALGOWiDDshliTd6wT99u0nCK8Z9+aozm
ut6Dacpps6kFtZaSF4fC0urQe87YQVt8rgIwRt7qy12a7DLCZRawTDBcMPPaTnOG
BtjOiQRINzf43TNRnXCve1XYAS59BWQOhriR

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRzCCAy+gAwIBAgIRAI4P+UuQcWhlM1T01EQ5t+AwDQYJKoZIhvcNAQELBQAww
PTELMaKGA1UEBhMCVVMxEjAQBgNVBAoTCUNlcnRhaW5seTEaMBGGA1UEAxMRQ2Vy
dGFpbmx5IFJvb3QgUjEwHhcNMjEwNDExMDAwMDAwWhcNNDYwNDExMDAwMDAwWjA9
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MRowGAYDVQQDEwFDZXXJ0
YWlubHkgUm9vdCBSMTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANA2
1B/q3avk0bbm+yLA3RMNansiExyXPGhjZjKcA7WNpIGD2ngwEc/csiu+kr+O5MQT
vqRoTNoCaBZ0vrLdBORrKt03H2As2/X3oXyVtwxwhi7xOu9S98zTm/mLvg7fMbed
aFySpvXl8wo0tf97ouSHocavFwDvA5HtqRxOcT3Si2yJ9HiG5mpJoM610rCrm/b0
1C7jcvk2xusVtyWMOvwlDbMicyF0yEqWYZL1LwsYpfSt4u5BvQF5+paMjRcCMLT5
r3gajLQ2EBAHBXDQ9DQGqilHFhiZ5shGIXsXwCITNSaa/ApzSRKft43jvRl5tcdF5
cBxGX1HpyTfcX35pe0HfNEXgO4T0oYoKNp43zGJS4YkNKPl6I7ENPT2a/Z2B7yyQ
wHtETrtJ4A5KVpK8y7XdeReJkd5hiXSSqOMyhb5OhaRLWcsrxXiOcVTQAjeZjOVJ
6uBUcQQRBi8LjMFbvrWhsFNunLhgkR9Za/kt9JQKl7XsxXYDVBtUrpMklZRNAB
2CnblJ2Oy0wQJuK0EJWtLeIAaSHO1OWzaMWj/Nmqhexx2DgwUMFDO6bW2BvBlyH
Wyf5QBGenDPBt+U1VwV/J84XIIwc/PH72jEpSe31C4SnT8H2TsIonPru4K8H+zMR
eiFPCyEQtkA6qyI6BjYlM4SGcprSp6XEtHWRqSsjAgMBAAGjQjBAMA4GA1UdDwEB
/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBbTgqj8ljZ9EXME66C6u
d0yEPmcM9DANBgkqhkiG9w0BAQsFAAOCAgEAuVevuBLaV4OPaAszHQNTVfSVcOQR
PbA56/qJYv331hgELyE03fFo8NWWwt7CgKPBjcZq91l3rhVxz1t5BXdm6ozTaw3d
8VksWTOlMIAVRQdFGjEitpIAq5lNOo93r6kiyi9jyhXWx8bwPWz8HA2YEGGeEali
1wrykXprOQ4vMMM2SZ/g6Q8CRFA3lFV96p/2O7qUpUzpvD5RtOjKkjZUbVwlKNrd
rRT90+7iIgXr0PK3aBLXWopBGsaSpVo7Y0VPv+E6dyIvXL9G+VoDhRNCX8reU9di
taY1BMJH/5n9hN9czulegChB8n3nHpDYT3Y+gjwN/KUD+nsa2UUeYNrEjvn8K8l7
lcUq/6qJ34IXD3L/DCfXCh5WAFaEJDJBlrXYFIW7pw0WwfgHJBu6haEaBQmAupVj
yTrsJZ9/nbqkRxWbRHDxakvWOF5D8xh+UG7pWijmZeZ3Gzr9Hb4DJqPb1OG7fpYn
Kx3upPvaJVQTA945xsMfTZDsxtK0hzthZU4UHIg1sGQUdGpXJpuHfUzVounmdLy
yCwzk5Iwx06MZTMQZBF9JBeW0Y3COMor6xOLRPIh80oat3df1+2IpHLIOR+Vnb5n
wXARPBv0+Em34yaXOp/SX3z7wJl8OSngex2/DaeP0ik0biQVY96QXr8axGbwua6
OV+KmalBWQewLK8=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDqDCCApCgAwIBAgIJAP7c4wEPyUj/MA0GCSqGSIb3DQEBBQUAMDQxCzAJBgNV
BAYTAkZSMRlWEAYDVQQKDAIEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hMB4X
DTA3MDYyOTE1MTMwNVVoXDTI3MDYyOTE1MTMwNVowNDELMAKGA1UEBhMCRIIEjA9
BgNVBAoMCURoaW5b3RpczERMA8GA1UEAwwIQ2VydGlnbmEwgEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDIaPHJ1tazNHUmgh7stL7qXOEm7RFHYeGifBZ4
QCHkYJ5ayGPhxLGWkv8YbWkj4Sti993iNi+RB7lIzw7sebYs5zRLcAglozyHGxny
gQcPOJAZ0xH+hrTy0V4eHpbNgGzOOzGTvKg0KmVEn2lmsxryIRWijOp5yIVUxbw
zBfsV1/pogqYCd7jX5xv3EjjhQsVWqa6n6xI4wmy9/Qy3l40vhx4XUJbzg4ij02Q
130yGLMLLGq/jj8UEYkgDncUtT2UCIf3JR7VsmAA7G8qKCvUk4YYxclPz5EIBb2
JsglrgVKtOdJLPOMFlN+XPGRGgjBRmKfIrjxwo1p3Po6WAbfAgMBAAGjgbwwgbkw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUGu3+QTmQtCRZvgHyUtVF9lo53BEw
ZAYDVROjBF0wW4AUGu3+QTmQtCRZvgHyUtVF9lo53BGhOKQ2MDQxCzAJBgNVBAYT
AkZSMRlWEAYDVQQKDAIEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hggkA/tzj
AQ/JSP8wDgYDVR0PAQH/BAQDAgEGMBEGCWCsAGG+EIBAQQEAwIABzANBgkqhkiG

9w0BAQUFAAOCAQEAhQMeknH2Qq/ho2Ge6/PAD/Kl1NqV5ta+aDY9fm4fTlrv0Q8h
bV6lUmPOEvjvKtpv6zf+EwLHyzs+ImvaYS5/1HI93TDhHkxAGYwP15zRgzB7mFnc
fca5DCIMoTOi62c6ZYTtLuLtdkVwj7Ur3vkj1kluPBS1xp81HIDQwY9qcEQCYsuu
HWhBp6pX6FOqB9IG9tUUBguRA3UsbHK1YZWaDYu5Def131TN3ubY1gkl2PlwS6w
t0QmwCbAr1UwnjvVNioZBPRcHv/PLLf/0P2HQBHVESO7SMAhqaQoLf0V+LBOK/Qw
WyH8EZE0vkHve52Xdf+XlcCWWC/qu0bXu+TZLg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGWzCCBEOgAwIBAgIRAMrpG4nxVQMNo+ZBbcTjpuEwDQYJKoZIhvcNAQELBQAw
WjELMAkGA1UEBhMCRIxEjAQBgNVBAoM CURoaW15b3RpczEcMBoGA1UECwwTMDDAw
MiA0ODE0NjMwODEwMDAzNjEzMBCGA1UEAwwQQ2VydGlnbmEgUm9vdCBDQTAeFw0x
MzEwMDEwODMyMjdaFw0zMzEwMDEwODMyMjdaMDFoxCzAJBgNVBAYTAkZSMRIwEAYD
VQKDAIEaGlteW90aXMxHDAaBgNVBASMeZAwMDIIGNDgxNDYzMDgxMDAwMzYxGTAX
BgNVBAMMEENlcnRpZ25hIFJvb3QgQ0EwggliMA0GCSCqGSIb3DQEBAQUAA4ICDwAw
ggIKAoICAQDNGDIlGlmx6mQWDoyUJJV8g9PFOSbcDO8WV43X2KyjQn+Cyu3NW9sO
ty3tRQgXstmzy9YXUUnIo245Onoq2C/mehJpNdt4iKVzSs9IGPjA5qXSjklYcoW9M
CiBtNyN6tMbaLOQdLNyzKNAT8kxOAkmhVECe5uUFoC2EyP+YbNDrihqECB63aCPu
I9Vwzm1RaRDuoXrC0SIxwoKF0vJVdlB8JXrJhFwLrN1CTivngqlkicuQstDuI7pm
TLtipPlTWmR7fJj6o0ieD5Wupxj0auwuA0Wv8HT4Ks16XdG+RCYyKfHx9WzMfglh
C59vpD++nVPiz32pLHxYGpfhPTc3GGY0o0kDFUYqMwy3OU4gkWGQwFsWq4NYKpkDf
ePb1BHxpE4S80dGnBs8B92jAqFe7OmGtBIyT46388NtEbVncSVmurJqZnJBBe3Yz
IoejwpKGBvlw7q6Hh5UbxHq9MfPU0uWZ/75I7HX1eBYdpnDBfzwb0ZL7z8g81sWT
Co/1VTp2lc5ZmIoJlXcymoO6LAQ6l73UL77XbJuiyn1tJslV1c/DeVIICZkHJC1k
JWumIWmbat10TWuXekG9qxf5kBDIjzb5LdXF2+6qhUVB+s06RbFo5jZMm5BX7CO5
hwjCxAnxl4YqKE3idMDaxIzb3+KhF1nOJfI0Mdp//TBt2dzhouH8XwIDAQABo4IB
GjCCARYwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYE
FBiHVuBud+4kNTxOc5of1uHieX4rMB8GA1UdIwQYMBaAFBiHVuBud+4kNTxOc5of
1uHieX4rMEQGA1UdIAQMDSwOQYEVROgADAxMC8GCCsGAQUFBwIBFiNodHRwcov
L3d3d3cuY2VydGlnbmEuZnIvYXV0b3JpdGVzLzBtBgNVHR8EZjBkMC+gLaArhilo
dHRwOi8vY3JsLmNlcnRpZ25hLmZyL2NlcnRpZ25hcm9vdGNhLmNybdAxoC+gLYYr
aHR0cDovL2Nybc5kaGlteW90aXMuY29tL2NlcnRpZ25hcm9vdGNhLmNybdANBgkq
hkiG9w0BAQsFAAOCAgEAlLieT/DjlQgi581oQfccVdV8AOItOoldaDgvUSILSo3L
6btdPrtcPbEo/uRTVRPPoZAbAh1fZkYJMyjhDSSXcNMQH+pkV5a7XdrnxIxpTGRG
HVyH41neQtGbqH6mid2PHMkwgu07nM3A6RngatgCdTer9zQoKJHyBApPNeNgJgH6
0BGM+RFq7q89w1DTj18zeTyGqHNfklwgtNjZfyO+B2XleJINugHA64wcZr+shncB
lA2c5uk5jR+mUYyZDDl34bSb+hxnV29qao6pK0xXeXpXIs/NX2NGjVxZOob4Mkdi
o2cNGJHc+6Zr9UhhcyNZjgKnvETq9Emd8VRY+WCV2hikLyhF3HqgiIZd8zvn/yk1
gPxxkQ5Tm4xxvvq0OKmOZK8l+hfZx6AYDlf7ej0gcWtSS6Cvu5zHbugRqh5jnxV/v
faci9wHYTfmJ0A6aBvmknpiZbyvKcL5kwIwJ9Omvw5Ip3IgWJJk8jSaYtlu3zM63
Nwf9JtmYhST/WSMDmu2dnajkXjjO11INb9I/bbEFa0nOipFGc/T2L/Coc3cOZayh
jWZSaX5LaAZHHjcn6WMxwLkFM1JABzS/3GkDpv0mztO+7skb6iQ12LAEpmJURw
3kAP+HwV96LOPNdeE4yBFxgX0b3xdx6A61GU5wSesVywlVP+i2k+KYTlerj1KjL0=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

CERTIFICATE-----

CERTIFICATE-----

b3JrIENBIDIwggIiMA0GCSqGSIlb3DQEBAQUAA4ICDwAwggIKAoICAQC9+Xj45tWA
DGSdhhuWZGc/IjoedQF97/tcZ4zJzFxrqZHmuULIIEub2pt7uZld2ZuAS9eEQCsn
0+i6MLs+CRqnSZXvK0AkwpfHp+6bJe+oCgCXhVqqndwpyeI1B+twTUrwWbNWuKFB
OJvR+zF/j+Bf4bE/D44WSWDXBo0Y+aomEKsq09DRZ40bRr5HMNUuctHFY9rnY3lE
fktjJlmgLjQ/KUxSiyqnwOKRKIm5wFv5HdnnJ63/mgKXwcZQkpsCLL2puTRZCr+E
Sv/f/rOf69me4Jgj7KZrdxYq28ytOxykh9xGc14ZYmhFV+SQgkK7QtbwYeDBoz1m
o130GO6IyY0XRSmZMnUCMe4pJshrAua1YkV/NxVaI2iJ1D7eTiew8EAMvE0Xy02i
sx7QBld9pPPV3WZ9fqGGmd4s7+W/jTcvedSVuWz5XV710GRBdxdaeOVDUO5/IOW
OZV7bIBaTxNyxtD9KXpEulKkKtVBRgkg/iKgtlswjbyJDNXXcPiHUv3a76xRLgez
Tv7QCdpw75j6VuZt27VXS9zllCUVyJ4ueE742pyehizKV/Ma5ciSixqClnrDvFAS
adgOWkaLOusm+iPjtrCBvkIApPjW/jAux9JG9uWOdf3yzLnQh1vMBhBgu4M1t15n
3kfsmUjxpKEV/q2MYo45VU85FrmxY53/twIDAQABo0IwQDAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBS2oVQ5AsOgP46KvPrU+Bym0ToO/TAOBgNVHQ8BAf8EBAMC
AQYwDQYJKoZIhvcNAQENBQADggIBAHHGIDs7k6b8/ONWJWsQCYftMxRQXLYtPU2sQ
F/xlhMcQSZDe28cmk4gmb3DWA145oPePq5a1pRNcgRRtDoGCERuKTSZPpd1iHkTf
CVn0W3cLN+mLIMb4Ck4uWBzrM9DPhmDJ2vuAL55MYIR4PSFk1vtBHxgP58l1cb29
XN40hz5BsA72udY/CROWFC/emh1auVbONTqwX3BNXuMp8SMoclm2q8KMZiYcdywm
djWLKKdpPk79SPdhrB0yZADVPHnr7pH1BKXESLjokmUbOe3lEu6LaTaM4tMpkT/
WjzGHWTYtTHkpx6qFcL2+1hGsvxznN3Y6SHb0xRONbkX8eftoEq5IVleVheO/jb
AoJnwTnbw3RLPTYe+SmTiGhbqEQZIfCn6IENLOiTNrQ3ssqwGyZ6miUfmpqAnksq
P/ujmv5zMnHCnsZy4YpoJ/HkD7TETKVhk/iXEAcqMCWpuchxuO9ozC1+9eB+D4Ko
b7a6bINDd82Kkhehnl4Fj1F4jNy3eFmynpTycUm/Q1oBEautmbjL4ZvrHG8hnj
XALKLNhvsGfZyTXaQHXYxKcZb55CEJh15pWLYLztxRLXis7VmFxFWlgPF7ncGNf/P
5O4/E2Hu29othfDNrp2yGAlFw5Khchf8R7agCyzxxN5DaAhqXzvwDMp7zAYspsbi
DrW5viSP
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFwDCCA6igAwIBAgIQHr9ZULjJgDdMBvfrVU+17TANBgkqhkiG9w0BAQ0FADB6
MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEgU3lzdGVtYBTLkEu
MScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlviBBdXR0b3JpdHkxH2AdBgNV
BAMTFkNlcnR1bSBUCnVzdGVkIFJvb3QgQ0EwHhcNMTgwMzE2MTIxMDEzWWhcNNDMw
MzE2MTIxMDEzWjB6MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEg
U3lzdGVtYBTLkEuMScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlmaWNhdGlviBBdXR0
b3JpdHkxH2AdBgNVBAMTFkNlcnR1bSBUCnVzdGVkIFJvb3QgQ0EwggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQDRLY67tzbqTeRn06TpwXkKQMIzhYc93yZ
n0EGZe2jusDbCSzBfN8pfktlL5On1AFrAygYo9idBcEq2EXxkd7fO9CAAozPOA/q
p1x4EaTByIVcJdPTsuclzxFUL6s1wB52HO8AU5853BSILCilS3Jy/I2z5T4IHhQq
NwuIPMqw9MjCoa68wb4pZ1Xi/K1ZXP69VyywkI3C7Te2fJmItDUDmj0VDT06qKhF
8JVOJVkdzZhpu9PMMsmN74H+rX2Ju7pgE8pllWeg8xn2A1bUatMn4qGtg/BKEiJ3
HAVz4hlxQsDsUaakFjgao4rpUYwBI4Zshfjvqm6f1bxJAPXSiEodg42MEx51UGa
mqi4NboMOvJEGyCI98U1z3G4z5D3Yf+xOr1Uz5MZf87Sst4WmsXXw3Hw09Omiqu
7VdNIuJGmj8PkTQkfVXjjJU30xrwCSss0smNtA0Aq2cpKNgB9RkEth2+dv5yXMSF
ytKAQd8FqKPVhJBPC/PgP5sZ0jeJP/J7UhyM9uH3PAeXjA6iWYEMspA90+NZRu0P
qafegGtaqge2Gcu8V/OXIXoMsSt0Puvap2ctTMSYnjYJdmZm/Bo/6khUHL4wvYBQ
v3y1zgD2DGHZ5yQD4OMBgQ692IU0iL2yNqh7XAJlRICMb/gv1SHKHRzQ+8S1h9E6
Tsd2tTVItQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBSM+xx1

vALTn04uSNn5YFSqxLNP+jAOBgNVHQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQENBQAD
ggIBAEEii1QALLtA/vBzVtVRJHlpr9OTy4EA34MwUe7nJ+jW1dReTagVphZzNTxl4
WxmB82M+w85bj/UvXgF2Ez8sALnNII5SW0ETsXpD4YN4fqzX4IS8TrOZgYkNCvo
zMnadyHncI013nR03e4qllY/p0m+jjGPP2Kh2RX5Rc64vmNueMzeMGQ2Ljdt4NR
5MTMI9UGfOZR0800McD2RrsLrfw9EAUqO0qRJe6M1ISHgCq8CYyqOhNf6DR5UMeq
GfnTKB7U0VEwKbOukGfWHwpjscWpxkIxYxeU72nLL/qMFH3EQxiJ2fAyQOaA4kZf
5ePBAFmo+eggvIksDkc0C+pXwlM2/KfUrzhN/gLldfq5Jwn58/U7yn2fqSLLiMmq
0Uc9NneoWWRrJ8/vJ8HjJLWG965+Mk2weWjROeiQWMOdvA8s1pfrzgzhIMfatz7D
P78v3DSk+yshzWePS/Tj6tQ/50+6uaWTRRxmHyH6ZF5v4HaUMst19W7l9o/HuKTM
qJZ9ZPskWkoDbGs4xugDQ5r3V7mzKWmTOPQD8rv7gmsHINF5H5pkAnuYZttcTVoP
0ISVoDwUQwbKytu4QTbaakRnh6+v40URFWkIsr4WOZckbxJF0WddCajJFdr60qZf
E2Efv4WstK2tBZQIgx51F9Nxo5NQI1mg7TyRVJ12AMXDuDjb

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICHTCCAaOgAwIBAgIUQ3CCd89NXTTxyq4yLzf39H91oJ4wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBGNVBAoMCUNvbW1TY29wZTERMCkGA1UEAwwiQ29t
bVNjb3BlIFB1YmtpYyBUcnVzdCBFQ0MgUm9vdC0wMTAeFw0yMTA0MjgxNzM1NDNa
Fw00NjA0MjgxNzM1NDJaME4xCzAJBgNVBAYTAiVtMRIwEAYDVQQKDAIDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDEw
djAQBgcqhkJOPQIBBgUrgQQAIGNiAARLNumuV16ocNfQj3Rid8NeeqrItqLxeP0C
flfdkXmcbLISiF58LwS+uM32ENEp7LXQoMPwiXAZu1FlxUOcW5tjnSCDPgYLpkJE
hRGnSjot6dZoL0hOUysHP029uax3OVejQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSOB2LAUN3GGQYARnQE9/OufXVNMDAKBggq
hkjOPQDAwNoADBlAjEAnDPfQeMjQEI2Jpc1XHvr20v4qotzVRVcrHgpD7oh2MSg
2NED3W3ROT3Ek2DS43KyAjB8xX6I01D1HiXo+k515liWpDVfG2XqYZpwI7UNo5uS
Um9poIyNStDuiw7LR47QjRE=

-----END CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIICHDCCAaOgAwIBAgIUkP2ZYEFHpgE6yhR7H+/5aAiDXX0wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBGNVBAoMCUNvbW1TY29wZTERMCkGA1UEAwwiQ29t
bVNjb3BlIFB1YmtpYyBUcnVzdCBFQ0MgUm9vdC0wMjAeFw0yMTA0MjgxNzQ0NTRa
Fw00NjA0MjgxNzQ0NTNaME4xCzAJBgNVBAYTAiVtMRIwEAYDVQQKDAIDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDIw
djAQBgcqhkJOPQIBBgUrgQQAIGNiAAR4MIHoYx7l63FRD/cHB8o5mXxO1Q/MMDAL
j2aTPs+9xYa9+bG3tD60B8jzlJHz7aRP+KNOjSkVWLjVb3/ubCK1sK9IRQq9qEmU
v4RDsNuESgMjGwdqb8FuvAY5N9GIveJQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBbTmGHX/72DehKT1RsfeSlXjMjZ59TAKBggq
hkjOPQDAwNnADbkAjAmc0l6tqvmSfr9Uj/UQQSugEODZXW5hYA4O9Zv5JOGq4/n
ich/m35rChJVYaoR4HkCMHfoMXGsPHED1oQmHhS48zs73u1Z/GtMMH9ZzkXpc2AV
mkzw5l4lIhVtwodZ0LKOag==

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFbDCCA1SgAwIBAgIUpgNjgXUWdDGOTKvVxZAplSUE5EN0wDQYJKoZIhvcNAQEL

BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCKGA1UEAwwi
Q29tbVNjb3BlIFB1YmxpYyBUcnVzdCBSU0EgUm9vdC0wMTAeFw0yMTA0MjgxNjQ1
NTRaFw00NjA0MjgxNjQ1NTNaME4xCzAJBgNVBAYTAIVTMRIwEAYDVQQKDAIDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUINBIFJvb3Qt
MDEwgglIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCwSGWjDR1C45FtnYSk
YZYSwu3D2iM0GXb26v1VWvZVAVMP8syMl0+5UMuzAURWlv2bKOx7dAvnQmtVzslh
suitQDy6uUEKBu8bJoWPQ7VAtYXR1HHcg0Hz9kXHgKKEUJdGzqAMxGBWBB0HW0al
DrJLpA6lf0741GIDuZNqihS4cPgugkY4lw50x2tBt9Apo52AsH53k2NC+zSDO3Oj
WiE260f6GBfZumbCk6SP/F2krfxQapWsvCQz0b2lf4b19bJzKo98rwjyGpg/qYFl
P8GMicWWMJoKz/TUyDTtnS+8jTiGU+6Xn6myY5QXjQ/cZip8UIF1y5mO6D1cv547
KI2DAg+pn3LiLCuz3GaXAEDQpFSOm117RTYm1nJD68/A6g3czhLmfTifBSeolz7p
UcZsBSjBAG/pGG3svZwG1KdJ9FQFa2ww8esD1eo9anbCyxooSU1/ZOD6K9pzig4H/
kQO9lLvkuI6cMmPNn7togbGEW682v3fuHX/3SZtS7NJ3Wn2RnU3COS3kuoL4b/JO
Hg9O5j9ZpSPcPYeoKFgo0fEbNttPxp/hjFtyjMcmAyejOQoBqsCyMWCDIqFPEgkB
Ea801M/XrmlTBQe0MXXgDW1XT2mH+VepuhX2yFJtocucH+X8eKg1mp9BFM6ltM6U
CBwJrVbl2rZJmkrqYxhTnCWuwwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUN12mmnQywsL5x6YVEFm45P3luG0wDQYJ
KoZlhcNAQELBQADggIBAK+nz97/4L1CjU3lIpbfaOp9TSp90K09FlxD533Ahuh6
NWPxzIHlxgvoLl1pKZJkGNRrDSsBTtXAOnTYtPZKdVUvhwQkZyybf5Z/Xn36lbQ
nmhUQo8mUuJM3y+Xpi/SB5io82BdS5pYV4jvguX6r2yBS5KPQJqTRlnLX3gWsWc+
QgvfKNmwrZggvkN80V4aCRckjXtdlemrwWCrWxhkgPut4AZ9HcpZuPN4KWfGVh2v
trV0Knahp/t1MJ+UXjulYPPLXAZiDslg+MkfFoom3ecnf+slpoq9uCO2EJqxWE2a
aE9gVOX2RhoOiKy8IUISrcZKiX2bwdgt6ZYD9KJ0DLwAHb/WNyVntHKLr4W96ioD
j8z7PEQkgulBpQtZtjSNMgsSDesnww1B10A8ckYpwlzqug/xBpMu95yo9GA+o/E4
Xo4TwbM6l4c/ksp4qRyv0LABJh6+cOx69TOY6lz/KwsETkPdY34Op054A5U+1C0w
lREQKC6/oAI+/15Z0wUOIv9TRe9rh9VlzRamloPh37MG88EU26fsHltdkJANclHn
YfkUyq+Dj7+vsQpZXdxcl+SWrVtgHdql7I52Qb1dgAT+GhMibA1xNxVssnBQVoc
icCMb3SgazNNtQEo/a2tiRc7ppqEvOuM6sRxJKi6KfkIsidWNTJf6jn7MZrVGczw
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFbDCCA1SgAwIBAgIUUVBa/O345lXGN0aoApYYNK496BU4wDQYJKoZIhvcNAQEL
BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTErMCKGA1UEAwwi
Q29tbVNjb3BlIFB1YmxpYyBUcnVzdCBSU0EgUm9vdC0wMjAeFw0yMTA0MjgxNzE2
NDNaFw00NjA0MjgxNzE2NDJaME4xCzAJBgNVBAYTAIVTMRIwEAYDVQQKDAIDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUINBIFJvb3Qt
MDIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDh+g77aAASyE3VrCLE
NQE7xVTIWXZjpX/rwcRqmL0yjReA61260WI9JSMZNRtpf4mnG2I81lDnNJUDMrG0
kyI9p+Kx7eZ7Ti6Hmw0zdQreqjXnfuU2mKKuJZ6VszKWpCtYHu8//mI0SFHrtI1C
rWDaSWqVcN3SAOLMV2MCe5bdSZdbkk6V0/nLKR8YSvgBKtJjCW4k6YnS5cciTNxz
hkcAqg2Ijq6FfUrpuzNPDlJwnZXjfG2WWy09X6GDRI224yW4fKcZgBzqZUPckXk2
LHR88mcGyYnJ27/aaL8j7dxrrSiDeS/sOKUNNWFnJ5rpM9kzXzehxfCrPfp4sOcs
n/Y+n2Dg70jpkEUeBVF4GiWSLFworA2iI540jwXmojPOEXcT1A6kHkIfhs1w/tku
FT0du7jyU1fbzMZ0KZwYszZ1OC4PVKH4kh+Jlk+71O6d6Ts2QrUKOyrUZHk2EOH5
kQMreyBUzQ0ZGshBMjTRsJnhkB4BQDa1t/qP5Xd1pCKBXbCL5CcSD1SIxtuFdOa3
wNemKfbr3vOTlycEVS8KbzfFPROvCgCpLlscgSjX74Yxqa7ybrjKaixUR9gqiC6v
wQcQeKwRoi9C8DfF8rhW3Q5iLc4Vn5V8qdE9isy9COoR+jUKgF4z2rDN6ieZdIs

5fq6M8EGRPbmz6UNp2YINios8wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUR9DnsSL/nSz12Vdgs7GxcJXvYXowDQYJ
KoZlhvcNAQELBQADggIBAIzpsU0v6Z9PIpNojuQhmaPORVMbc0RTAIFhzTHjCLqB
KCh6krm2qMhDnscTJk3C2OVVnJJdUNjCK9v+5qiXz1I6JMNIZFxHMaNINRPDK7n3
+VGXu6TwYof1gbTl4MgqX67tiHCpQ2EAOHyJxCDut0DgdXdaMNmEMjRdrSzbyme
APnCKfWxkxlSaRosTKCL4BWaMS/TiJVZbuXEs1DIFAhKm4sTg7GkcrI7djNB3Nyq
pgdvHSQSn8h2vS/ZjvQs7rfSOBAkNIEv41xdgSGn2rtO/+YHqP65DSdsu3BaVXoT
6fEqSWnHX4dXTEN5bTpl6TBcQe7rd6VzEojov32u5cSoHw2OHG1QAk8mGEPej1WF
sQs3BWDJVTkSBKEqz3EWnzZRSb9wO55nnPt7eck5HHisd5FUmrh1CoFSI+NmYWvt
PjgelmFV4ZFUjO2MJB+ByRCac5krFk5yAD9UG/iNuovnFNa2RU9g7Jauwy8CTl2d
lklyALKrdVwPaFsdZcJfMw8eD/A7hvWwTruc9+olBdytoptLFwG+Qt81IR2tq670
v64fG9PiO/yzcNMcmyiQiRM9HcEARwmWmjgb3bHPDcK0RPOWlc4yOo80nOAXx17O
rg3bhjlp1v9mxnhMUF6cKojawHhRUzNlM47ni3niAli9G7oyOzWPPO5std3eqx7

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIC2zCCAmCgAwIBAgIQfMmPK4TX3+oPyWWa00tNljAKBggqhkJOPQQDAzBIMQsw
CQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQQDExELVRS
VVNUIEJSIFJvb3QgQ0EgMSAyMDIwMB4XDTIwMDIxMTA5NDUwMFOxDTM1MDIxMTA5
NDQ1OVowSDELMakGA1UEBhMCREUxFTATBgNVBAoTDEQQtVHJ1c3QgR21iSDEiMCAG
A1UEAxMZRC1UUIVTVCBBCUiBSb290IENBIDEgMjAyMDIwMDIwMDIwMDIwMDIwMDIw
BAAIAiA2IABMBLxyjR+4T1mu9CFCdHq2tuda38KwOE1HaTJddZO0Flax7mNCq7dPYS
zuht56vkPE4/RAiLzRZxy7+SmfSk1zxQVFKQhYN4lGdnoxwJGT11NIXe7WB9xwy0
QVK5buXuQqOCAQ0wggEJMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBByEFHOREKv/
VbNafAkl1bK6CKBrqx9tMA4GA1UdDwEB/wQEAwIBBjCBxgYDVR0fBIG+MIG7MD6g
PKA6hjhodHRwOi8vY3JsLmQtdHJ1c3QubmV0L2Nybc9kLXRydXN0X2JyX3Jvb3Rf
Y2FfMV8yMDIwLmNybdB5oHegdYzZbGRhcDovL2RpcmVjdG9yeS5kLXRydXN0Lm5l
dC9DTj1ELVRSVVNUJTIwQlIlMjBSb290JTIwQ0ElMjAxJTIwMjAyMCxPPUQtVHJ1
c3QlMjBHbWJILEM9REU/Y2VydGhmaWNhdGVyZXZvY2F0aW9ubGlzdDAKBggqhkJO
PQQDAwNpADBmAjEAIJAte/rhY/hhY+ithXhUkZy4kzg+GkHaQBZTQgjKL47xPoFW
wKrY7RjEsK70PvomAjEA8yjixtsrmfu3Ubgko6SUeho/5jbiA1czijDLgsfWFBHV
dWNbFJWcHwHP2NVypw87

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFqTCCA5GgAwIBAgIQczswBEhb2U14LnNLyaHcZjANBgkqhkiG9w0BAQ0FADBI
MQswCQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAYDVQQQDExELVRS
LVRSVVNUIEJSIFJvb3QgQ0EgMiAyMDIzMB4XDTIzMDUwOTA4NTYzMVoXDTM4MDUw
OTA4NTYzMFOwSDELMakGA1UEBhMCREUxFTATBgNVBAoTDEQQtVHJ1c3QgR21iSDEi
MCAGA1UEAxMZRC1UUIVTVCBBCUiBSb290IENBIDIGMjAyMzCCAIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgCggIBAK7/CVmRgApKaOYkP7in5Mg6CjoWzckjYaCTcfKr
i3OPoGdlYNJUa2NRb0kz4HIHE304zQaSBYlSa053bATTlfrdTIZXcFhfUvnKLNE
gXtRr90zsWh81k5M/itoucpmacTsXld/9w3HnDY25QdgrMBM6ghs7wZ8T1soegj8
k12b9py0i4a6Ibn08OhZWiihNIQaJZG2tY/vsvmA+vk9PBFy2OMvhnbfFeSzBqZCT
Rphny4NqoFAjpvz2gTng7fC5v2Xx2Mt6++9zA84A9H3X4F07ZrjcjrDy4d2A/wl
2ecjbwb9Z/Pg/4S8R7+1FhhGaRTMBff00msa8yr5LULQyReS2tNZ9/WtT5PeB+U

cSTq3nD88ZP+npNa5JRal1QMNXtfbO4AHyTsA7oC9Xb0n9Sa7YUsOCIvx9gvdhFP
/Wxc6PWOJ4d/GUohR5AdeY0cW/jPSoXk7bNbjb7EZChdQcRurDhaTyN0dKkSw/bS
uREVMweR2Ds3OmMwBtHfJjYoYiMQ4EbMl6zWK11kJNXuHA7e+whadSr2Y23OC0K+
0bpwHJwh5Q8xaRfX/Aq03u2AnMuStIv13lmiWAmly0cL4UEyNEHZmrHZqLabWt4N
DfTisl01gLmB1IRpkQLLddCNxbU9CZEJjXShFHR5PtbJFR2kWVki3PaKRT08EtY+
XTIvAgMBAAGjgY4wgYswDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUZ5Dw1t61
GNVGKX5cq/ieCLxklRAwDgYDVR0PAQH/BAQDAgEGMEkGA1UdHwRCMEAwPqA8oDqG
OGh0dHA6Ly9jcmwuZC10cnVzdC5uZXQvY3JsL2QtdHJ1c3RfYnJfc9vdF9jYV8y
XzIwMjMuY3JsMA0GCSqGSIb3DQEBDQUAA4ICAQA097N3U9swFrktpSHxQCF16+tI
FoE9c+CeJyrrd6kTpGoKWloUMz1oH4Guaf2Mn2VsNELZLdB/eBaxOqwJMa1ef67n
riv6uvw8l5VAk1/DLQOj7aRvU9f6QA4w9QAglABMjDu0ox+2v5Eyq6+SmNMW5tTR
VFxDWy6u71cqqlRvpO8NVhTalasgdp4D/Ca4nj8+AybmTNudX0KEPUUDAXXZiMrc
LmEkWqTqJwzEr5SswrPMhfiHocaFpVibVrg0M8JkiZmkdijYQ6qgYF/6FKC0ULn
4B0Y+qSFNueG4A3rvNTJ1jxD8V1Jbn6Bm2m1iWKPifLY1/4nwSPFyysCu7Ff/vtD
hQNGv13GyiEm/9cCnnRK3PgTfBGBVzbLZVzRHTF36SXDw7IyN9XxmAnkbWOACKsG
koHU6XCPpz+y7YaMgmo1yEJagtFSGkUPFaUA8JR7ZSdXOUPPFH/mvTWze/EZTN46
ls/pdu4D58JDUjxqgejBWOC9EV2Ta/vH5mQ/u2kc6d0li690yVRAysuTEwrt+2aS
Ecr1wPrYg1UDfNPfIkZ1cGt5SAYqgpg/5usWDiJfAbzdNpQ0qTUmiteXue4Icr80
knCDgKs4qll03UCkGJCy89UDyibK79XH4I9TjvAA46jtn/mtd+ArY0+ew+43u3gJ
hJ65bvspmZDogNOFJA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIC2zCCAmCgAwIBAgIQXwJB13qHfEwDo6yWjfv/0DAKBggqhkJOPQQDAzBIMQsw
CQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAyDVQQDExlELVRS
VVNUIEVWIFJvb3QgQ0EgMSAyMDIwMB4XDTIwMDIxMTEwMDAwMFOxDTM1MDIxMTA5
NTk1OVowSDELMakGA1UEBhMCREUxFTATBgNVBAoTDEQQtVHJ1c3QgR21iSDEiMCAG
A1UEAxMZRC1UUIVTVCBFViBSb290IENBIDEgMjAyMDIwMDIwMDAwMFOxDTM1MDIxMTA5
BAAiA2IABPEL3YZDIBnfl4XoIkqbz52Yv7QFJsnL46bSj8WeeHsxiamJrSc8ZRCC
/N/DnU7wMyPE0jL1HLDfMxddxfCxivnvubcUyilKwg+pf3VlSSowZ/Rk99Yad9rD
wpdhQntJraOCAQ0wggEJMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFH8QARY3
OqQo5FD4pPfsazK2/umLMA4GA1UdDwEB/wQEAwIBBjCBxgYDVR0fBIG+MIG7MD6g
PKA6hjhodHRwOi8vY3JsLmQtdHJ1c3QubmV0L2Nybc9kLXRydXN0X2V2X3Jvb3Rf
Y2FfMV8yMDIwLmNybdB5oHegdYzZbGRhcDovL2RpcmVjdG9yeS5kLXRydXN0Lm5l
dC9DTj1ELVRSVVNUJTIwRVYIMjBs290JTIwQ0ElMjAxJTIwMjAyMDIwMDIwMDAwMFOxDTM1MDIxMTA5
c3QlMjBhWJILEM9REU/Y2VydGlmaWNhdGVyZXZvY2F0aW9ubGlzdDAKBggqhkJOP
PQQDAwNpADBmAjEAYjzGKnXCXnViOTYAYFqLwZOZzNnbQTS7h5kXO9XMT8oi96CA
y/m0sRtW9XLS/BnRAjEakfcwkz8QRitxpNA7RJvAKQIFskF3UfN5Wp6OFKBOQtJb
gfM0agPnIjhQW+0ZT0MW

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFqTCCA5GgAwIBAgIQaSYJfoBLTKCnjHhiU19abzANBgkqhkiG9w0BAQ0FADBI
MQswCQYDVQQGEwJERTEVMBMGA1UEChMMRC1UcnVzdCBHbWJIMSIwIAyDVQQDExlELVRS
VVNUIEVWIFJvb3QgQ0EgMiAyMDIwMB4XDTIwMDIxMTEwMDAwMFOxDTM1MDIxMTA5
OTA5MTAzMlowSDELMakGA1UEBhMCREUxFTATBgNVBAoTDEQQtVHJ1c3QgR21iSDEi

MCAGA1UEAxMZRC1UUIVTVCBFViBSb290IENBIDIgMjAyMzCCAiIwDQYJKoZIhvcN
AQEBBQADggIPADCCAgCggIBANiOo4mAC7JXUtpU0w3uX9jFxPvp1sjW21sJkK
F8GLxNuo4MwxusLyzV3pt/gdr2rEIYfXR8mV2IIEUD2BCP/kPbOx1sWy/YgJ25yE
7CUXFId/MHibaljJtnMoPDT3mfd/06b4HEV8rSyMID/YZxBTfiLNTiVR8CUkNRFe
EMbsh2aJgWi6zCudR3Mfvc2RpHJqnKibGKBv7FD0fUDCqDDPvXPIEysQEx6Lmqg6
IHPTGGkKSv/BAQP/eX+1SH977ugpbzZMIWGG2Pmic4ruri+W7mjNPU0oQvIFKzIb
RIUWaqZLKfm7lVa/Rh3sHZMdwGWyH6FDrlaeoLGPaxK3YG14C8qKXO0elg6DpkiV
jTujIcSuWMYAsoS0I6SWHjW42J7YrDRJmGOVxcttSEfi8i4YHtAxq9107PncjLgc
jmgjutDzUNzPZY9zOjLHfP7KgiJPvo5iR2blzYfi6NUPGJ/IBHJLRjwQ8kTCZFZx
TnXonMkmdMV9WdEKWw9t/p51HBjGGjp82A0EzM23RWV6sY+4roRIPrN6TagD4uJ+
ARZzaBhDM7DS3LAaQzXupdqRlyuhoFBAUp0JuyfBr/CBTdkdXgpaP3F9ev+R/nk
hbDhezGdpn9yo7nELC7MmVcOIQxFAZRl62UJxmMiCzNJkkg8/M3OsD6Onov4/knF
NXJHAgMBAAGjgY4wgYswDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUqvyREBuH
kV8Wub9PS5FeAByxMoAwDgYDVR0PAQH/BAQDAgEGMEkGA1UdHwRCMEAwPqA8oDqG
OGh0dHA6Ly9jcmwuZC10cnVzdC5uZXQvY3JsL2QtdHJ1c3RfZXZfcm9vdF9jYV8y
XzIwMjY3JmMA0GCSqGSIb3DQEBDQUAA4ICAQCTy6UfmRHsmg1fLBWTxj++EI14
QvBukEdHjqOSMo1wj/Zbjb6JzkcBahsgIllbyIIQbODnmaprxigYzWRaoUlrRc4
pZt+UPJ26oUFKIDBK7GB0aL2QHWpDsvxVUjY7NHss+jOFKE17MJeNRqrphYBB07q
3C+jisoksetSjl8MmxfPy3MHGcRqwnNU73xDUmPBEcrCRbH0O1P1aa4846XerOhU
t7KR/aypH/KH5BfGSah82ApB9PI+53c0BFLd6IHYS9URZ0V4U/M5d40VxDJI3IX
cI1QcB9WbMy5/zpaT2N6w25lBx2Eof+pDGOJbbJAiDnXH3dotfyc1dZnaVuodNv8
ifYbMvekJKZ2t0dT741Jj6m2g1qlpBFYfXeA08mD6iL8AOWsKwV0HFaanuU5nCT
2vFp4LJiTZ6P/4mdm13NRemUAiKN4DV/6PEEeXfSVIP4M7kFMhtYVRFP0OUr3Hs
7dpn1mKmS00PaaLJvOwiS5THaJQXfuKOKD62xur1NGyfn4gHONuGcfrNIUhDbqNP
gofXNjhuS5N5YHVPd/Aa1VP6IQzCP+k/HxiMkl14p3ZnGbuy6n/pcAlWVqOwDAst
Nl7F6cTVg8uGF5csbBNvh1qvSaYd2804BC5f4ko1Di1L+KIkBI3Y4WNeApI02pPh
XBxvWHZks/wCuPWdCg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIEMzCCAxugAwIBAgIDCYPzMA0GCSqGSIb3DQEBDQUAAQME0xCzAJBgNVBAYTAkRF
MRUwEwYDVQQKDAxELVRydXN0IEdtYkgxJzAlBgNVBAMMHkQtVFJlU1QgUm9vdCBD
bGFzcyAzIENBIDIgMjAwOTAeFw0wOTExMDUwODM1NThaFw0yOTExMDUwODM1NTha
ME0xCzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEdtYkgxJzAlBgNVBAMM
HkQtVFJlU1QgUm9vdCBDDbGFzcyAzIENBIDIgMjAwOTCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBAN0ySs96R+91myP6Oi/WUEWJNTrGa9v+2wBoqOADER03
UAifTUpolDwzU9GUY6cgVq/eUXjsKj3zSEhQPgrfRIWLJ23DEE0NkVJD2IfgXU42
tSHKXzlABF9bfsyjiupQB7ZNoTWSPSHjRGICTBpFGOShrvUD9pXRI/RcPHAY9R
ySPocq60vFYJfxLLHLGvKZAKyVXMD900Gu1HNVpK7ZxzBCHQqr0ME7UAyiZsxGsM
lFqVlNpQmvH/pStmMaTJOKDfHR+4CS7zp+hnUquVH+BGptikw8paxTGA6Eian5Rp
/hnd2HN8gcqW3o7tszIFZYQ05ub9VxClX3a/L7AQDcUCAwEAAaOCARowggEWMA8G
A1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFp3aFMSfMN4hvr5COfyryYnJ4PGEMA4G
A1UdDwEB/wQEAwIBBjCB0wYDVR0fBIHLMiHIMIGA0H6gfIZ6bGRhcDovL2RpcmVj
dG9yeS5kLXRydXN0Lm5ldC9DTj1ELVRSVVNUJTIwUm9vdCUyMENsYXNzJTlwMyUy
MENBJTIwMiUyMDIwMDksTz1ELVRydXN0JTlwR21iSCxDPURFP2NlcnRpZmljYXRl
cmV2b2NhdGlvbmxc3QwQ6BBoD+GPWh0dHA6Ly93d3cuZC10cnVzdC5uZXQvY3Js
L2QtdHJ1c3Rfcm9vdF9jbGFzc18zX2NhXzJfMjAwOS5jcmwwDQYJKoZIhvcNAQEL

BQADggEBAH+X2zDI36ScfSF6gHDOFBjpiBSVYEQBrLLpME+bUMJm2H6NMLVwMeni
acfczNsgFYbQDfC+rAF1hM5+n02/t2A7nPPKHeJeaNijnZflQGDSNiH+0LS4F9p0
o3/U37CYAqxva2ssJSRyoWXuJVrl5jLn8t+rSfrzkGkj2wTZ51xY/GXU177M/C4K
zCUqNQ4YJEVdT1B/yMfGchs64JTBKbkTCJNjYy6zltz7GRUUG3RnFX7acM2w4y8
PIWmawomDeCTmGCufsYkl4phX5GOZpIJhzbNi5stPvZR1FDUWSi9g/LMKHtThm3Y
Johw1+qRzT65ysCQblRGXnRl11z+o+I=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIEQzCCAyugAwIBAgIDCYP0MA0GCSqGSIb3DQEBCwUAMFAxCzAJBgNVBAYTAkRF
MRUwEwYDVQQKDAxELVRydXN0IEEdtYkgxKjAoBgNVBAMMIUQtVFJlVU1QgUm9vdCBD
bGFzcyAzIENBIDIGRVYgMjAwOTAeFw0wOTExMDUwODUwNDZaFw0yOTExMDUwODUw
NDZaMFAxCzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEEdtYkgxKjAoBgNV
BAMMIUQtVFJlVU1QgUm9vdCBDDbGFzcyAzIENBIDIGRVYgMjAwOTCCASlWdQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAJnxhDRwui+3MKCOvXwEz75ivJn9gpfSegpn
ljgJ9hBOISJzmY3aFS3nBfwZcyK3jpgAvDw9rKFs+9Z5JUut8Mxk2og+KbgPCdM0
3TP1YtHhzRnp7hhPTFiu4h7WDFsVWtg6uMQYZB7jm7K1iXdODL/ZlGsTl28So/6Z
qQTMFexgaDbtCHu39b+T7WYxg4zGcTSHThfqr4uRjRxWQa4iN1438h3Z0S0NL2IR
p75mpoo6Kr3HGrHhFPC+Oh25z1uxav60sUYgovseO3Dvk5h9jHOW8sXvhXCtKSb8
HgQ+HKDYD8tSg2J87otTlZCpV6LqYQXY+U3EJ/pure3511H3a6UCAwEAAaOCASQw
ggEgMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFNOUikxiEyoZLsyvcp9Ntea
HNxnMA4GA1UdDwEB/wQEAwIBBjCB3QYDVROfBIHVMIHSMIGHoIGEoIGBhn9sZGFw
Oi8vZGlyZWN0b3J5LmQtdHJ1c3QubmV0L0NOPUQtVFJlVU1QIMjBSb290JTlwQ2xh
c3MlMjAjJTlwQ0ElMjAyJTlwRVYlMjAyMDA5LE89RC1UcnVzdCUyMEdtYkgsQz1E
RT9jZXJ0aWZpY2F0ZXJldm9jYXRpb25saXN0MEagRKBChkBodHRwOi8vd3d3LmQt
dHJ1c3QubmV0L2NyY3k5LXRydXN0X3Jvb3RfY2xhc3NfM19jYV8yX2V2XzlwMDku
Y3J5MA0GCSqGSIb3DQEBCwUAA4IBAQA07XtaPKSUiO8aEXUHL7P+PPoeUSbrh/Yp
3uDx1MYkCenBz1UbtDDZzhr+BlGmFaQt77JLvyAoJUnRpjZ3NOhk31KxEcdzes05
nsKtjHEh8lpr988TIWvsoRlFIm5d8sqMb7Po23Pb0iUMkZv53GMOkaEGTcH8gNF
CSuGdXzfX2lXANtu2KZyIktQ1HWYVt+3GP9DQ1CuekR78HIR10M9p9OB0/DJT7na
xpeG0ILD5EJt/rDiZE4OJudANCa1CInXCGNjOCd1HjPqbqjdn5lPdE2BiYBL3ZqX
KVwvvoFBuYz/6n1gBp7N1z3TLqMVvKjmJuVvw9y4AyHqnxbxLFS1
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDtzCCAp+gAwIBAgIQDOfg5RfYRv6P5WD8G/AwOTANBgkqhkiG9w0BAQUFADBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJvb3QgQ0EwHhcNMDYxMTEwMDAwMDAwWWhcNMZExMTEwMDAwMDAwWjBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJvb3QgQ0EwEggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCtDhXO5EOAXLGH87dg+XESpa7cJpSIqvTO9SA5KFhgDPiA2qkVITJhPLWxKISKityfCgyDF3qPkKyK53lTXDGEKvYPmDI2dsze3Tyoo9q+yHyUmHfnyDXH+Kx2f4YZNISW1/5WBg1vEfNoTb5a3/UsDg+wRvDjDPZ2C8Y/igPs6eD1sNuRMBhNZYW/lmci3Zt1/GiSw0r/wty2p5g0I6QNcZ4VYcgoc/lbQrISXwxmDNslumH0DJaoroTghHtORedmTpyoeb6pNnVFzF1roV9Iq4/

AUaG9ih5yLHa5FcXxH4cDrC0kqZW572yl+2qp/C3xag/lRbQ/6GW6whfGHdPAgMB
AAGjYzBhMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBRF66Kv9JLLgJEtUYunpyGd823IDzAfBgNVHSMEGDAWgBRF66Kv9JLLgJEtUYun
pyGd823IDzANBgkqhkiG9w0BAQUFAAOCAQEAog683+Lt8ONyc3pkL/3cmbYMuRC
dWKuh+vy1dneVrOfzM4UKLkNl2BcEkxY5NM9g0lFWJc1aRqoR+pWxnmrEthngYTf
fwk8lOa4JiwgvT2zKIn3X/8i4peEH+ll74fg38FnSbNd67lJKusm7Xi+ft8r87cm
NW1fiQG2SVufAQWbqz0lwcy2f8Lxb4bG+mRo64EtI0tCt/qMHt1i8b5QZ7dsvfPx
H2sMNgcWfzd8qVttevESRmCD1ycEvkvOl77DZypoEd+A5wwzZr8TDRRu838fYxAe
+o0bJW1sj6W3YQGx0qMmoRBxna3iw/nDmVG3KwcIzi7mULKn+gpFL6Lw8g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDljCCAn6gAwIBAgIQC5McOtY5Z+pnI7/Dr5r0SzANBgkqhkiG9w0BAQsFADBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkieIEIFJvb3QgRzIwHhcNMTIwMDAwWjBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkieIEIFJvb3QgRzIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZ5ygvUj82ckmIkzTz+GoeMVSA
n61UQbVH35ao1K+ALbkKz3X9iaV9JPrJlgwrvJUXCzO/GU1BBpAAvQxNEP4Htecc
biJVMWWXvdMX0h5i89vqbFCMP4QMls+3ywPgym2hFEwbid3tALBSfK+RbLE4E9Hp
EgjAALAcKxHad3A2m67OeYfcgnDmCXRwVWmvo2ifv922ebPynXApVfSr/5Vh88lA
bx3RvpO704gqu52/clpWcTs/1PPRCv4o76Pu2ZmvA9OPYLfykqGxvYmJHzDNw6Yu
YjOuFgJ3RfrngQo8p0Quebg/BLxcoIfhG69Rjs3sLPr4/m3wOnyqi+RnlTGNAgMB
AAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQDAgGGMB0GA1UdDgQW
BBTOW0q5mVXyuNtgV6l+vVa1lzan1jANBgkqhkiG9w0BAQsFAAOCAQEAyqVVjOPI
QW5pJ6d1Ee88hjZv0p3GeDgdaZaikmkuOGybfQTUiaWxMTeKySHMq2zNixya1r9I
0jJmwYrA8y8678Dj1JGG0VDjA9tZd29KOVpt3ibHtX2vK0LRdWLjSisCx1BL4Gni
lmwORGYQRI+tBev4eaymG+g3NJ1TyWGqolKvSnAWhsI6yLETcDbYz+70CjTVW0z9
B5yiutkBclzzTcHdDrEcDcRjvq30FPuJ7KJBDkzMyFdA0G4Dqs0MjomZmWzWPDcV
ON9vvKO+KSAnq3T/EyJ43pdSVR6DtVQgA+6uwE9W3jfMw3+qBCe703e4YtsXfJwo
IhNzbM8m9Yop5w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICRjCCAc2gAwIBAgIQC6Fa+h3foLVJRK/NJKBs7DAKBggqhkiOPQQDAzBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkieIEIFJvb3QgRzIwHhcNMTIwMDAwWjBIMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkieIEIFJvb3QgRzIwMwdjAQBgcqhkjOPQIBBgUrgQQAIGNiAAQZ57ysRGXtzbG/WPuNsVepRC0FFfLvC/8QdJ+1YlJfZn4f5dwbRXkLzMZTCp2NXQLZqVneAlr2lSoOjThKiknGvMYDOAdfVdp+CW7if17QRSAPWXYQ1qAk8C3eNvJsKTmjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQDAgGGMB0GA1UdDgQWBBTL0L2p4ZgFUaFNN6KDec6NHSrkhDAKBggqhkiOPQQDAwNnADBkAjAlpIFFAmsSS3V0T8gj43DydXLeFIwz5FyYZ5eEJJZVrmDxxDnOOlY

JjZ91eQ0hjkCMHw2U/Aw5WJjOpnitqM7mzT6HtoQknFekRON3aRukswy1vUhZscv

6pZjamVFkpUBtA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDrzCCApegAwIBAgIQCDvgVpBCRRrGhdWrrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDEXdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAEFw0wNjExMTAwMDAwMDBaFw0zMTEwMTAwMDAwMDBaMGExCzAJBgNVBAYTAIVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RlZ2lDZXJ0IEZsbnB290IENBMIIIBjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTTo1eqUKKPC3eQyaKl7hLollsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5clzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkiYIvUX7Q6hL+hqkpMfT7P
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSsiicNoxBN33shbyTAPOB6jtSj1etX+jkMOvJwIDAQABO2MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR
TLtm8KPiGxvDI7I90VUwHwYDVR0jBBgwFoAUA95QNVbR TLtm8KPiGxvDI7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkloyQY/Esr
hMATudXH/vTBH1jLuG2cenTnmCmrEbXjKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jpmP6P6fbtGbfYmbW0W5Bjflttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBlqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls
YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4d3p30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrciXpg0X40KPMbp1ZWVbd4=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDjjCCAnagAwIBAgIQAzrx5qcRqAC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDEXdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAEFw0xMzA4MDEwMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAIVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RlZ2lDZXJ0IEZsbnB290IEcyMIIIBjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNNx7a8myaJCtSnX/RrohCgiN9RIUyfuI
2/Ou8jqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FR1IAWsULecYxpsMNzaHxmx
1x7e/dfgy5SDN67sH0NO3Xss0r0upS/kqbitOtSZpLYl6ZtrAGCSYP9PIUkY92eQ
q2EGnI/yuum06ZIya7XzV+hdG82MHauVBJVJ8zUtlUNJbd134/tJS7SsVQepj5Wz
tCO7TG1F8PapsPUwtP1MVYwnSlcUfIKdzXOS0xZKBgyMUNGPHgm+F6HmIcr9g+UQ
vIOlCsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMWxM4MphQIDAQABO0IwQDAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUtIUIBiV
5uNu5g/6+rkS7QYXjzkWdQYJKoZIhvcNAQELBQADggEBAGBnKJRvDkhj6zHd6mcY
1Yl9PMWLSn/pvtsrF9+wX3N3KjITOYFnQoQj8kVnNeyIv/iPsGEMNKSulEyExtv4
NeF22d+mQrvHRAiGfzZ0JFrabA0UWTW98kndth/Jsw1HKj2ZL7tcu7XUIOGZX1NG
Fdtom/DzMNU+MeKNhJ7jitalj41E6Vf8PlwUHBHQRFXGU7Aj64GxJUTFy8bJZ91
8rGOmaFvE7FBcf6IKshPECBV1/MURexGRPTqh5Uykw7+U0b6LJ3/iyK5S9kJRaTe

pLiaWN0bfVKfjllDiIGknibVb63dDcY3fe0Dkhvld1927jyNxF1WW6LZZm6zNTfl

MrY=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICPzCCAcWgAwIBAgIQBVVWvPJepDU1w6QP1atFcjAKBggqhkhjOPQQDAzBhMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cu
ZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBHMzAe
Fw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAiVTMRUw
EwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5jb20x
IDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSc290IEczMHYwEAYHKoZIzj0CAQYF
K4EEACIDYgAE3afZu4q4C/sLfyHS8L6+c/MzXRq8NOrexp80JX28MzQC7phW1FG
fp4tn+6OYwwX7Adw9c+ELkCDnOg/QW07rdOkFFk2eJ0DQ+4QE2xy3q6Ip6FrtUPO
Z9wj/wMco+I+o0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAWIBhjAd
BgNVHQ4EFgQU9tIpPmhxdIuNkHMEWNpYim8S8YwCgYIKoZIzj0EAwMDAAAwZQIx
AK288mw/EkrRLTnDCgmXc/SINoyIJ7vmil1Qhadj+Z4y3maTD/HMsQmP3Wyr+mt/
oAIwOWZbwmSNuJ5Q3KjVSalTx9zRSX8XAbjIho9OjIgrqJqpisXRAL34VOKa5Vt8
sycX

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDxTCCAq2gAwIBAgIQAxqCJmoLQJuPC3nyrkYldzANBgkqhkiG9w0BAQUFADBs
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSswKQYDVQQDEyJEaWdpQ2VydCBlaWdoIEFzc3VyYW5j
ZSBFViBSc290IENBMjB4XDTA2MTEwMDAwMDAwMFoXDTMxMTEwMDAwMDAwMFowbDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ2lDZXJ0IEluYzEzMjBcGA1UECmMQd3d3
LmRpZ2ljZXJ0LmNvbTErMCKGA1UEAxMiMRGlnaUNlcnQgSGlnaCBBC3N1cmFuY2Ug
RVYgUm9vdCBDQTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMbM5XPm
+9S75S0tMqbf5YE/yc0lSbZxKsPVIDRnogocsF9ppkCxxLeyj9CYpKIBWTrT3JTW
PNt0OKRkZEOlgvdKpVMSOO7zSW1xkX5jtqumX8OkhPhPYIG++MXs2ziS4wblCJEM
xChBVfvLWokVfnHoNb9Ncgk9vjo4Uft3MRuNs8ckRZqnrG0AFFoEt7oT61EKmEFB
Ik5IYYeBQVCmeVyJ3hlKV9Uu5l0cUyx+mM0aBhakaHPQNAQTXXFx01p8VdteZOE3
hzBWBOUrtCmAeVf5OYiiAhF8J2a3iLd48soKqDirCmTCv2ZdlYTBoSUeh10aUAsg
EsxBu24LUTi4S8sCAwEAANjMGExDgYDVR0PAAQH/BAQDAgGGMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBByEFLE+w2kD+L9HAdSYJhoIAu9jZCvDMB8GA1UdIwQYMBaA
FLE+w2kD+L9HAdSYJhoIAu9jZCvDMA0GCSqGSIb3DQEBBQUAA4IBAQAAGgaX3Nec
nzyIZgYIVyHbIUf4KmeqvxygdkAQV8GK83rZEWwONfqe/EW1ntlMMUu4kehDLI6z
eM7b41N5cdblIZQB2lWHmiRk9opmzN6cN82oNLFpmyPinngiK3BD41VHMWEZ71jF
hS9OMPagMRYjyOfiZRYzy78aG6A9+MpeizGLYAiJLQwGXFK3xPkKmNEVX58Svnw2
Yzi9RKR/5CYrCsSXaQ3pjOLAEFe4yHYSkVXySGnYvCoCWw9E1CAx2/S6cCZdkGCe
vEsXCS+0yx5DaMkHJ8HSXPfqIbloEpw8nL+e/IBcm2PN7EeqJSdnoDfzAIJ9VNep
+OkuE6N36B9K

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICGTCCAZ+gAwIBAgIQCeCTZaz32ci5PhwLBCou8zAKBggqhkhjOPQQDAzBOMQsw

CQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQsIEluYy4xJjAkBgNVBAMTHURp
Z2lDZXJ0IFRmUyBFQ0MgUDM4NCBSb290IEc1MB4XDTIxMDExNTAwMDAwMFoXDTQ2
MDExNDIzNTk1OVowTjELMAkGA1UEBhMCMVVMxZzAVBgNVBAoTDkRlPzZlDZXJ0LCBJ
bmMuMSYwJAYDVQQDEx1EaWdpQ2VydCBUTFMgRUNDIFAzODQgUm9vdCBHNTB2MBAG
ByqGSM49AgEGBSuBBAAiA2IABMFEoc8Rl1Ca3iOCNqfN0MsYndLxf3c1TzvdIHJS
7cI7+Oz6e2tYIOyZrsn8aLN1udsJ7MgT9U7GCh1mMEy7H0cKPGEQQil8pQgO4CLp
0zVozptjn4S1mU1YoI71VOeVyaNCMEAwhQYDVR0OBByEFMFRRVBZqz7nLFr6ICIS
B4CifBFqMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MAoGCCqGSM49
BAMDA2gAMGUCMQCJao1H5+z8blUD2WdsJk6Dxv3J+ysTvLd6jLRl0mlpYxNjOyZQ
LgGheQaRnUi/wr4CMEfDFXuxoJGZSZOoPHzoRgaLLPIxAJSdYsiJvRmEFOMl+wG4
DXZDjC5Ty3zfDBeWUA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFZjCCA06gAwIBAgIQCPm0eKj6ftpqMzeJ3nzPijANBgkqhkiG9w0BAQwFADBN
MQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQsIEluYy4xJTAjBgNVBAMT
HERpZ2lDZXJ0IFRmUyBSU0E0MDk2IFJvb3QgRzUwHhcNMjEwMTE1MDAwMDAwWhcN
NDYwMTE0MjM1OTU5WjBNMQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQs
IEluYy4xJTAjBgNVBAMTHERpZ2lDZXJ0IFRmUyBSU0E0MDk2IFJvb3QgRzUwWggli
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCz0PTJeRGd/fxmgefM1eS87IE+
ajWOLrnf3q/5B03PMJ3qCQuZvWxX2hhKuHisOjmopkisLnLlvevxGs3npAOPxG0
2C+JFvuUAT27L/gTBaF4HI4o4EXgg/RZG5Wzrn4DReW+wkL+7vI8toUTmDKdFqgp
wgscONyFMXdcvyey/Cestyu9dJsXLfKB2l2w4SMXPohKEiPQ6s+d3gMXsUJKoBZM
pG2T6T867jp8nVid9E6P/DsjyG244gXazOvswzH016cpVIDPRFtMbZCe88zdH5RD
nU1/cHAN1DrRN/BsnZvAFJNY781BOHW8EwOVfH/jXOnVDdXifBBiqmvwPXbzP6Po
sMH976pXTayGpxi0KcEsDr9kvimM2AltVwv8n/vFfQMfawKsPHTDU9qTXeXAaDx
Zre3zu/O7Oyldcqs4+Fj97ihBMi8ez9dLRYiVu1ISf6nL3kwJZu6ay0/nTvEF+cd
Lvvyz6b84xQslpghjLSR6Rlgg/IwKwZzUNWYOWbpx4oMYIwo+FKbbuH2TbsGJJvX
KyY//SovcfXWJL5/MZ4PbeiPT02jP/816t9JXkGPhvnx3d3Lg7SjXi/7RgLQZhNe
XoVPzthwiHvOAbWWl9fNff2C+MikwcoBOU+NosEUQB+cZtUMCUbW8tDRSHZWokPL
tgoRObqME2wGtZ7P6wIDAQABo0IwQDAdBgNVHQ4EFgQUUTMc7TZArxFTJc1paPKv
TiM+s0EwDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcN
AQEMBAQDggIBAGCmr1tfV9qJ20tQqcQjNSH/0GEwhJG3PxDPJY7Jv0Y02cEhJhxxw
GXleo8mH/qldZJY6yFMECrZBu8RHANmfGBg7sg7zNOok992vIGCukihfNudd5N7H
PNtQOa27PShNlnx2xlv0wdsUpasZYgcYQF+Xkdycx6u1UQ3maVNVzDI92sURVXLf
O4uJ+DQtpBfIF+aZfTCiITfNMBc9uPK8qHWgQ9w+iUuQrm0D4ByjoJYJu32jtyoQ
REtGBzRj7TG5BO6jm5qu5jF49OokYTurWGT/u4cnYiWB39yhL/btp/96j1EuMPik
AdKFOV8BmZzVWltwGUb+hmA+rYAQCd05JS9Yf7vSdPD3Rh9GOUrYU9DzLjtxpdRv
/PNn5AeP3SYZ4Y1b+qOTEZvpyDrDVWiakuFSdjj04bq9+0/V77PnSIMx8IIh47a+
p6tv75/ftM8BuGJqIz3nCU2AG3swpMPdB380vqQmsvZB6Akd4yCYjqdP//fx4ilw
MUc/dNAUFvohigLVigmUdy7yWSiLffCSCmZ4OIN1xLVaqBHG5cGdZlXPU8Sv13WF
qUITVuwhd4GTWgzqltJyqEI8pc7bZsEGCREjnwB8twl2F6GmrE52/WRMmrRpnCK
ovfepEWFJqgejF0pW8hL2JpqA15w8oVPbEtoL8pU9ozaMv7Da4M/OMZ+
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

HQBUIwbgsQfZxw9cZX08bVlX5O2ljelAU58VS6Bx9hoh49pwBiFYFIeFd3mqgnkC
AwEAAaNCMEAwHQYDVR0OBBYEFOLJQJ9NzuiaoXzPDj9lxSmIahlRMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEBCwUAA4ICAQDRSVfg
p8xoWLoBDysZzY2wYUWsEe1jUGn4H3++Fo/9nesLqjJHdtJnJO29fDMylrHBYZm
DRd9FBUb1Ov9H5r2XpdptxolpAqzkT9fNqyL7FeoPueBihhXOYV0GkLH6VsTX4/5
C0mSdl31R9KrO9b7eGZONn356ZLpBN79SWP8bfsUcZNnL0dKt7n/HipzcEYwv1ry
L3ml4Y0M2fmyYzeMN2WFcGpcWwlyua1jPLHd+PwyvzeG5LuOmCd+uh8W4XAR8gPf
JWlyJyYYMoSf/wA6E7qaTfRPuBRwIrHKK5DOKcFw9C+df/KQHtZa37dG/OaG+svg
IHZ6uqbL9XzeYqWxi+7egmaKTjowHz+Ay60nugxe19CxVsp3cbK1daFQqUBDF8Io
2c9Si1vIY9RCPqAzekYu9wogRIR+ak8x8YF+QnQ4ZXMn7sZ8uI7XpTrXmKGcjBBV
09tL7ECQ8s1uV9JiDnxXk7Gnbc2dg7sq5+W2O3FYrf3RRbxake5TFW/TRQl1brqQ
XR4EzfffHqhmsYzmlGrv/EhOdJhCrylvLmrH+33RZjEizIYAfmaDDEL0vTSSwxrq
T8p+ck0LcIymSLumoRT2+1hEmRSuqguTaaApJUqlyyvdimYHFngVV3Eb7PVHhPOe
MTd61X8kreS8/f3MboPoDKi3QWwH3b08hpcv0g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgjCCA2qgAwIBAgILWku9WvtPilv6ZeUwDQYJKoZIhvcNAQELBQAwwTTELMakG
A1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9uaXRvcmluZyBHbWJIMRkw
FwYDVQQDExBHTE9CQUxUUIVTVCAyMDIwMB4XDTIwMDIxMDAwMDAwMFoXDTQwMDYx
MDAwMDAwMFowTTELMakGA1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9u
aXRvcmluZyBHbWJIMRkwFwYDVQQDExBHTE9CQUxUUIVTVCAyMDIwMIIICjANBgkq
hkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAri5WrRsc7/aVj6B3GyvTY4+ETUWiD59b
RatZe1E0+eyLinjF3WuvvcTfk0Uev5E4C64OFudBc/jbu9G4UeDLgztzOG53ig9Z
YybNpyrOVPU44sB8R85gfD+yc/LAGbaKkoc1DZAoouQVBGM+uq/ufF7MpotQsjj3
QWPKzv9pj2gOITblzLmMCcpL3TGQlsjMH/1WljTbjhzqLL6FLmPdqqmV0/0plRPw
yJiT2S0WR5ARg6I6IqIoV6Lr/sCMKKCMfecqJquCgGOIYx8ZzHyyZqjC0203b+J+
BIHZRYQfEs4kUmSFC0iAToexIiIwquuuuuAC4EDosEKAA1GqtH6qRNdDYfOiaxaJ
SaSjpCuKAsR49GiKweR6NrFvG5Ybd0mN1MkGco/PU+PcF4UgStyYJ9ORJitHHmkH
r96i5OTUawuzXnzUJIBHKWk7buis/UDr2O1xcSvy6Fgd60GXIsUf1DnQJ4+H4xj0
4KlGdFv0OoIu0G4skaMxXDtG6nsEEFZegB31pWXogvziB4xiRfUg3kZwhqG8k9Me
dKZssCz3AwyIDMvUclOGvGBG85hqwvG/Q/lwIHfKN0F5VVJjVsSn8VoxIldrPIw
q7ejMZdnrY8XD2zHc+0klGvIg5rQmjdBKuxFshsSUktq6HQJLYqUp5ISXbY9e2
nKd+Qmn7OmMCAwEAAANjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVR0OBBYEFNwuH9FhN3nkq9XVsxJxaD1qaJwiMB8GA1UdIwQYMBaAFNwu
H9FhN3nkq9XVsxJxaD1qaJwiMA0GCSqGSIb3DQEBCwUAA4ICAQCR8EICaEDuw2JA
VC/f7GLDw56KoDEoqoOOpFaWEhCGVrqXctJUMHyGdUdaG/7FELYjQ7ztdG14wJC
XtzoRlghHNQIw4Lx0SsFDKv/bGtCwr2zD/cuz9X9tAy5ZVp0tLTWMstZDFyySCstd
6IwPS3BD0IL/qMy/pJTAvoe9iuOTE8aPmxadJ2W8esVCgmxcB9CpwYhgROmYhRZf
+I/KARDOJcP5YBugxZfD0yyIMaK9MOzQ0MAS8cE54+X1+NZK3TTN+2/BT+MAi1bi
kvcoskJ3ciNnxz8RFbLEAwW+uxF7Cr+obuf/WEPPm2eggAe2HcqtbePBEX4tdJP7
wry+UUTF72glJ4DjyKDUEuzZpTcdN3y0kcralLGWge9oXHYQSa9+pTeAsRxSvTOB
TI/53WXZFM2KJVj04sWDpQmQ1GwUY7VA3+vA/MRYfg0UFodUJ25W5HCEuGwyEn6C
MUO+1918oa2u1qsgEu8KwxCMSZY13At1XrFP1U80DhEgB3VDRemjEdqso5nCtnkn
4rnvyOL2NSl6dPrFf4IFYqYK6miyeUcGbvJXqBUzxvd4Sj1Ce2t+/vdG6tHrju+I
aFvowdlxfv1k7/9nR4hYJS8+hge9+6jlgqispdNpQ80xiEmEU5LAsTkBOYMBMMTy
qfrQA71yN2BWHZz8vTmR9W0Nv3vXkg==

-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFVzCCAz+gAwIBAgINAgPlk28xsBNJiGuiFzANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEU
MBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZp
Y2VzIEExMQZEUUMBIGA1UEAxMLR1RTIFJvb3QgUjEwggLiMA0GCSqGSIsb3DQEBAQUA
A4ICDwAwggIKAoICAQC2EQKLHuOhd5s73L+UPreVp0A8of2C+X0yBoJx9vaMf/vo
27xqLpeXo4xL+Sv2sfnohB2x+cWX3u+58qPpvBKJXqeqUqv4IyflPLGcY9vXmX7w
Cl7raKb0xlpHDU0QM+NOsROjyBhsS+z8CZDfnWQpJSMHobTSPS5g4M/SCYe7zUjw
TcLceoiKu7rPWRnWr4+wB7CeMfGCwcDfLqZtbBkOtdh+JhpFAz2weaSUKK0Pfybl
qAj+lug8aJRT7oM6iCsVlgmy4HqMLnXWnOunVmSPlk9orj2XwoSPwLxAwAtcvfaH
szVsrBhQf4TgTM2S0yDpM7xSma8ytSmzJSq0SPly4cpk9+aCEI3oncKKiPo4Zor8
Y/kB+Xj9e1x3+naH+uzfsQ55IVe0vSbv1gHR6xYKu44LtcXFilWr06zqkUspzBmk
MiVOKvFIRNACzqrOSbTqn3yDsEB750Orp2yjj32JgfpMpf/VjsPOS+C12LOORc92
wO1AK/1TD7Cn1TsNsYqiA94xrcx36m97PtbkSIS5r762DL8EGMUUXLeXdYWk70p
aDPvOmbB4om3xPXV2V4J95eSRQAogB/mqghtqmxlbCluQ0WEdrHbEg8QOB+DVrN
VjzRlW5y0vtOUucxD/SVRNuJLDWcfr0wbrM7Rv1/oFB2ACYPTIrImqYNxgFIQID
AQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4E
FgQU5K8rJnEaK0gnhS9SZizv8IkTcT4wDQYJKoZIhvcNAQEMBQADggIBAJ+qQibb
C5u+/x6Wki4+omVKapi6Ist9wTrYggoGxval3sBOh2Z5ofmmWJyq+bXmYOfg6LEe
QkEzCzc9zolzFcq1JKjPa7XSQCGYzyI0zzvFIoTgxQ6KfF2I5DUkzps+GIQebtuy
h6f88/qBVRRIcImpIgUxPoLW7ttXNLwzldMXG+gnoot7TiYaelpktGsN/H9oPM4
7HLwEXWdyzRSjeZ2axfG34arJ45JK3VmgRAhpua+9K4l/3wV3s6MJT/KYnAK9y8J
ZgflPxz88NtFMN9iiMG1D53Dn0reWVIHxYciNuaCp+0KueIHol17eko8cdLiA6Ef
MgfG+RCzgwARWGAtQsgWSl4vflVy2PFPEz0tv/bal8xa5meLMFrUKTX5hgUvYU/
Z6tGn6D/Qqc6f1zLXbBwHSs09dR2CQzreExZBfMzQsNhFRAbd03OIozUhfJFbdT
6u9AWpKQXCbfTkBdYiJ23//OYb2MI3jSNwLgjt7RETeJ9r/tSQdirpLsQBqvFAnZ
0E6yove+7u7Y/9waLd64NnHi/Hm3lCXRSNboTXns5lndcEZoitHTtNCjv0xyBZm
2tIMPNUzjsmhDYAPexZ3FL//2wmUspO8IFgV6dtxQ/PeEMMA3KgqlbbC1j+Qa3bb
bP6MvPJwNQzcmRk13NfIRmPVNnGuV/u3gm3c

-----END
CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIIFVzCCAz+gAwIBAgINAgPlrsWNBCUaqxElqjANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEU
MBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZp
Y2VzIEExMQZEUUMBIGA1UEAxMLR1RTIFJvb3QgUjEwggLiMA0GCSqGSIsb3DQEBAQUA
A4ICDwAwggIKAoICAQDO3v2m++zsFDQ8BwZabFn3GTxd98GdVarTzTukk3LvCvpt
nfbwhYBboUhSnznFt+4orO/LdmgUud+tAWyZH8QiHZ/+cnfgLFuv5AS/T3KgGjSY
6Dlo7JUle3ah5mm5hRm9iYz+re026nO8/4Piy33B0s5Ks40FnotJk9/BW9BuXvAu
MC6C/Pq8tBcKSOWIm8Wba96wyrQD8Nr0kLhlZPdCTK3ofmZemde4wj7I0BODre7k
RXuJVfeKH2JShBKzwcCX44ofR5GmdFrS+LFjKBC4swm4VndAoiaYecb+3yXuPuWg

f9RhD1FLPD+M2uFwdNjCaKH5wQzpoeJ/u1U8dgbuak7MkogwTZq9TwtImoS1mKPV
+3PBV2HdKFZ1E66HjucMUQkQdYhMvI35ezzUIkgfKtzra7tEscszcTJGr61K8Yzo
dDqs5xoic4DSMPclQsciOzsSrZYuxsN2B6ogtzVJV+mSSeh2FnIxZyuWfojqx5RW
Ir9qS34BIbljMt/kmkRtWVtd9QCgHJvGeJeNkP+byKq0rxFROV7Z+2et1VsRnTKa
G73VululycslaVNVJ1zgyjbLiGH7HrfQy+4W+9OmTN6SpdT3/UGVN4unUu0kzCq
gc7dGtxRcw1PcOnlthYhGXmy5okLdWTK1au8CcEYof/UVKGFPP0UJA0yh9OktwID
AQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4E
FgQUu//KjiOfT5nK2+JopqUVJxce2Q4wDQYJKoZIhvcNAQEMBQADggIBAB/Kzt3H
vqGf2SdMC9wXmBFqiN495nFWcrKeGk6c1SuYJF2ba3uwM4IJvd8lRuqYnrYb/oM8
0mJhwQTtzuDFycgTE1XnqGOTjHsB/ncw4c5omwX4Eu55MaBBRTUoCnGkJE+M3DyC
B19m3H0Q/gxhswWV7uGugQ+o+MePTagjAiZrHYNSVc61LwDKgEDg4XSsYPWHgJ2u
NmSRXbBoGOqKYcl3qJfEycel/FVL8/B/uWU9J2jQzGv6U53hkRrJXRqWbTKH7QMg
yALOWr7Z6v2yTcQvG99fevX4i8buMTolUVVnjWQye+mew4K6Ki3pHrTgSAai/Gev
HyICc/sgCq+dVeuHzf9gR7A/Xe8bVr2XIZYtCtFenTgCR2y59PYjJbigapordwj6
xLEokCZYCDzifqrXPW+6MYgKBesntaFJ7qBFVHvmJ2WZICGoo7z7GJa7Um8M7YNR
TOI24iBgxcJlkoKM8xAfDoqXvneCbT+PHV28SSe9zE8P4c52hgQjxcCMElv924Sg
JPFI/2R80L5cFtHvma3AH/vLrrw4IgyMZNralw4/KBVEqE8AyyCazM90arQ+POuV
7LXTWtiBmelDGDfrs7rVWGJB82bSj6p4IVQgw1oudCvV0b4YacCs1aTPObpRhANl
6WLAYv7YTVWW4tAR+kg0Eeye7QUd5MjWHYbL

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCTCCAY6gAwIBAgINAgPluLrIPglJ209ZjAKBggqhkJOPQQDAzBHMqSwCQYD
VQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEUMBIG
A1UEAxMLR1RTIFJvb3QgUjMwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMqSwCQYDVQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2Vz
IEExMQZEUMBIGA1UEAxMLR1RTIFJvb3QgUjMwdjAQBgcqhkJOPQIBBgUrgQQAIGNi
AAQfTzOHMymKoYTey8chWEGJ6ladK0uFxb1MJ7x/JIFyb+Kf1qPKzEUURout736G
jOyxfi//qXGdGIRFBEFVbivqJn+7kAHjSxm65FSWRQmx1WyRRK2EE46ajA2ADDL2
4CejQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBTB8Sa6oC2uhYHP0/EqEr24Cmf9vDAKBggqhkJOPQQDAwNpADBMAjEA9uEglRR7
VKOQFhG/hMjqb2sXnh5GmCCbn9MN2azTL818+FsuVbu/3ZL3pAzCMeGiAIEA/Jdm
ZuVDFhOD3cfl74UOO0BzrEXGhF16b0DjyZ+hOXJYKaV11RZt+cRLInUue4X

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCTCCAY6gAwIBAgINAgPlwGjvYxqccpBQUjAKBggqhkJOPQQDAzBHMqSwCQYD
VQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEUMBIG
A1UEAxMLR1RTIFJvb3QgUjMwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMqSwCQYDVQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2Vz
IEExMQZEUMBIGA1UEAxMLR1RTIFJvb3QgUjMwdjAQBgcqhkJOPQIBBgUrgQQAIGNi
AATzdHOnaItgrkO4NcWBMHtLSZ37wWHO5t5GvWvVYRg1rkDdc/eJkTBa6zzuhXyi
QHY7qca4R9gq55KRanPpsXI5nymfopjTX15YhmUPoYRlBtHci8nHc8iMai/lxKvR
HYqjQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBSATNbrdP9JNqPV2Py1PsVq8JQdjDAKBggqhkJOPQQDAwNpADBMAjEA6ED/g94D
9J+uHXqnLrmvT/aDHQ4thQEd0dlq7A/Cr8deVl5c1RxYIigL9zC2L7F8AjEA8GE8

p/SgguMh1YQdc4acLa/KNJvxn7kjNuK8YAOdGLoAVsjh4rsUecrNIdSUtUID

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIB3DCCAYOgAwIBAgINAgPlfvU/k/2lCSGypjAKBggqhkhjOPQQDAjBQMSQwIgYD
VQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0gUjQxEzARBgNVBAoTCkdsb2Jh
bFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTIxMTEzMDAwMDAwWhcNMzgw
MTE5MDMxNDA3WjBQMSQwIgYDVQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0g
UjQxEzARBgNVBAoTCkdsb2JhbFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wWTAT
BgqhkhjOPQIBBgqhkhjOPQMBBwNCAAS4xnnTj2wlDp8uORkcA6SumuU5BwkWymOx
uYb4ilfBV85C+nOh92VC/x7BALJucw7/xyHlGKSq2XE/qNS5zowdo0IwQDAOBNV
HQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUUVLB7rUW44kB/
+wpu+74zyTjyhNUwCgYIKoZIZj0EAwIDRWAwRAIglk90crlgr/HmnKAWBVBfw147
bmF0774BxL4YSFlhgjICICadVGNA3jdgUM/I2O2dgq43mLyjj0xMqTQrbO/7lZsm

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICHjCCAAsgAwIBAgIRYFIJ4CYuu1X5CneKcflK2GwwCgYIKoZIZj0EAwMwUDEk
MCIGA1UECxMbR2xvYmFsU2lnbiBFQ0MgUm9vdCBDQSA1FI1MRMwEQYDVQKEwpH
bG9iYWxTaWduMRMwEQYDVQDEwPHbG9iYWxTaWduMB4XDTEyMTEzMDAwMDAwMFoX
DTM4MDExOTAzMTQwN1owUDEkMCIGA1UECxMbR2xvYmFsU2lnbiBFQ0MgUm9vdCBD
QSA1FI1MRMwEQYDVQKEwpHbG9iYWxTaWduMRMwEQYDVQDEwPHbG9iYWxTaWdu
MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAER0UOlvt9Xb/pOdEh+J8LttV7HpI6SFkc
8GIxLcB6KP4ap1yztSY50XUWPrRd21DosCHZTQKH3rd6zWzocWdTARvQZU4f8ke
hOvRnkmSh5SHDDqFSmafNvmTTZdhBoZKo0IwQDAOBNVHQ8BAf8EBAMCAYYwDwYD
VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUPEYpSJvqB8ohREom3m7e0oPQn1kwCgYI
KoZIZj0EAwMDaAAwZQIXAOVpEsLU28YxuglB4Zf4+/2a4n0Sye18ZNPLBSWLvtmg
515dTguDnFt2KaAJJiFqYgIwcdK1j1zqO+F4CYWodZI7yFz9SO8NdCKoCOJuxUnO
xwy8p2Fp8fc74SrL+SvzZpA3

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDXzCCAkegAwIBAgILBAAAAAABIVhTCKIwDQYJKoZIhvcNAQELBQAwTDEgMB4G
A1UECxMXR2xvYmFsU2lnbiBSb290IENBIC0gUjMxEzARBgNVBAoTCkdsb2JhbFNp
Z24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMDkwMzE4MTAwMDAwWhcNMjkwMzE4
MTAwMDAwWjBMMSAwHgYDVQQLExtHbG9iYWxTaWduIFJvb3QgQ0EgLSBSMzZETMBEG
A1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMwldpB5BngiFvXAg7aEyiie/QV2EcWtiHL8
RgJDx7KKnQRfJMsuS+FggkbhUqsMgUdwbN1k0ev1LKMPgj0MK66X17YUhhB5uzsT
gHeMCOFJ0mpiLx9e+pZo34knltifBtc+ycsmWQ1z3rDI6SYOgxXG71uL0gRgykmm
KPZpO/bLyCiR5Z2KYVc3rHQU3HTgOu5yLy6c+9C7v/U9AOEGM+iCK65TpjoWc4zd
QQ4gOsC0p6Hpsk+QLjJg6VfLuQSSaGjLOCZgdbKfd/+RFO+uIE8rUAVSNECMWEZ
XriX7613tSaer9fwRPvm2L7DWzgVGkWqQPabumDk3F2xmmFghcCAwEAAaNCMEAw
DgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFI/wS3+o
LkUkrk1Q+mOai97i3Ru8MA0GCSqGSIb3DQEBCwUAA4IBAQBQLQNVaUKr+yAzv95ZU
RUm7lgAJQayzE4aGKAzymvmdLm6AC2upArT9fHxD4q/c2dKg8dEe3jgr25sbwMp

jjM5RcOO5LlXbKr8EpbsU8Yt5CRsuZRj+9xTaGdWPoO4zzUhw8lo/s7awlOqzJCK
6fBdRoyV3XpYKBovHd7NADdBj+1EbddTKJd+82cEHhXXipa0095MJ6RMG3NzdvQX
mcIfeg7jLQitChws/zyrVQ4PkX4268NXSb7hLi18YIvDQVETI53O9zJrlAGomecs
Mx86OyXShkDOOyyGeMlhLxS67ttVb9+E7gUJTb0o2HLO02JQZR7rkpeDMdmztcph
WD9f

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgzCCA2ugAwIBAgIORea7A4Mzw4VlSOB/RVEwDQYJKoZIhvcNAQEMBQAwtDEg
MB4GA1UECXMxR2xvYmFsU2lnbiBSb290IENBIC0gUjYxZzARBgNVBAoTCkdsb2Jh
bFNpZ24xZzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTQxMjEwMDAwMDAwWhcNMzQx
MjEwMDAwMDAwWjBMMSAwHgYDVQQLExdHbG9iYWxTaWduIFJvb3QgQ0EgLSBSNjET
MBEGA1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgoCggIBAJUH6HPKZvnsFMp7PPcNCPG0RQssgrRI
xutbPK6DuEGSMxSkb3/pKszGsIhrxbaJ0cay/xTOURQh7ErdG1rG1ofuTTToVBu1k
ZguSgMpE3nOUTvOniX9PeGMIyBJQbUJmL025eShNUhqKGoC3GYEOfsSKvGRMIRxD
aNc9PIrFsmBvKjQ3MQbFvuJtMgamHvm566qjuL++gmNQ0PAYid/kD3n16qIfKtJw
LnnvJO7bVPiSHyMEAc4/2ayd2F+4OqMPKq0pPbzUoSB239jLKJz9CgYXfIWHSw
1CM69106yqLbnQneXUQtKPGBzVeS+n68UARjNN9rkxi+azayOeSsJDa38O+2HBNX
k7besvjhbzdorg1qkXy4J02oW9UivFyVm4uiMVRQkQVlO6jxTiWm05OWgtH8wY2
SXcwwHE35absIQh1/OZhFj931dmRl4QKbNQCTXTAFO39OfuD8l4UoQSwC+n+7o/h
bguyCLNhZglqsQY6ZZZZwPA1/cnaKI0aEYdwgQqomnUdnjqGBQCe24DWJfncBZ4n
WUx2OVvq+aWh2IMP0f/fMBH5hc8zSPXKbWQULHpYT9NLCEnFlWQaYw55PfWzjMpY
rZxCRXluDocZXFSSxZba/jJvcE+kNb7gu3GduyYsRtYQUigAZcIN5kZeR1Bonvzce
MgfYFGM8KEyvAgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBSUBAWjKxPioufi1xzWx/B/yGdToDAfBgNVHSMEGDAWGBSu
bAWjKxPioufi1xzWx/B/yGdToDANBgkqhkiG9w0BAQwFAAOCAGEAgyXt6NH9lVLN
nsAEoJFp5lzQhN7craJP6Ed41mWYqVuoPI8AorRbrcWc+ZfwFSY1XS+wc3iEZGt
Ixxg93eFyRJa0lV7Ae46ZeBZDE1ZXs6KzO7V33EByrKPrnzU+sQghoefEQzd5Mr61
55wsTLxDKZmOMNOsleDjHfrYBzN2VAAiKrlNIC5waNrlU/yDXNOd8v9EDERm8tLj
vUYAGm0CuiVdjaExUd1URhxN25mW7xocBFymFe944Hn+Xds+qkxV/ZoVqW/hpvpf
cDDpw+5CRu3CkwWJ+n1jez/QcYF8AOiYrg54NMML+68KnyBr3TsTjxKM4kEaSHpz
oHdpX7Zcf4LIHv5YGygrqGytXm3ABdJ7t+uA/iU3/gKbaKxCXcPu9czc8FB10jZp
nOZ7BN9uBmm23goJSFmH63sUYHpkqmlD75HHTOwY3WzvUy2MmeFe8nl+z1TivWfs
pA9MRf/TuTAjB0yPEL+GltmZWrsZVxykzLsViVO6LAUP5MSeGbEYNNVMnbrt9x+v
JJUEeKgDu+6B5dpffltKoZB0JaezPkvILFa9x8jvOOJckvB595yEunQtYQEgfn7R
8k8HWV+LLUNS60YMI0H1Zkd5d9VUWx+tJDfLRVpOoERiyNiwmcUVhAn21klJwGW4
5hpxbqCo8YLoRT5s1gLXCmeDBVrJpBA=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCzCAZGgAwIBAgISEdK7ujNu1LzmJGjFDYQdmOhDMAoGCCqGSM49BAMDMEYx
CzAJBgNVBAYTAkFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYDVQQD
ExNHbG9iYWxTaWduIFJvb3QgRTQ2MB4XDTE5MDMyMDAwMDAwMFoXDTE5MDMyMDAw
MDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2Ex
HDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBFNDYwdjAQBgcqhkiOPQIBBgUrgQQA

IgNiAAScDrHPt+ieUnd1NPqlRqetMhkytAepJ8qUuwzSChDH2omwlwxwEwkBjtjq
R+q+soArzfwoDdusvKSGN+1wCAB16pMLEy5SnCNoIwZD7JlvU4Tb+0cUB+hflGdd
yXqBPCCjQJBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1Ud
DgQWBbQxCpCptsad0kRLgLiWi5h+xEk8blTAKBggqhkhjOPQQDAwNoADBlAjEA31SQ
7Zvvi5QCkxeCmb6zniz2C5GMn0oUsfZkvLtoURMMA/cVi4RguYv/Uo7njLwcAjA8
+RHUjE7AwWHCFUyqqx0LMV87HOIA0Qx5v5zli/altP+CAezNI8BZ/3Hobui3A=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBAUA
MEYxChAJBgNVBAYTAkFMRkwFwYDVQQKEExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDEeXNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwgglMA0GCSqGSIb3DQEBA
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNI5brtQyYdpokNRbopiLkKHPd08EsCveJ
OaFv6Wc0dwxu5FUdUiXSE2te4R2pt32Jm8Nnp8semNgQB+msLZ4j5lUghYruQG
vGIFAha/r6gjA7aUD7xubMLL1aa7DOn2wQL7ld5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHif8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+Ctl4wSE
y132tGqzZfxCnlEmIyDLPRT5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412IPFzYE
+cCQYDdF3uYM2HSNrpYibXRdQr4G9dlkgbIqRImwTDsHTUB+JMWKmIJ5jqSngiCN
I/onccnfxkF0oE32kRbcRoxfKWMxWxEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvqgo6GDoLclcqUC
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLltbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFxzHmuEajwfmf3lH7wvqMxX63bElazHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczzF2sUyQQCPtlk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQRzRQDGQob4Ez8pn7fXwgNNgyYmQlgXQBztSvwyeqiv5
u+YfjyW6hY0XHgI+XVAEV8/+LbZvXMAaq7afJMbc2hIkCwU9D9SGuTSyXTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzmE6JwYZ+vj+vYxXX4M2bUdGc6
N3ec592kD3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgXT7PntgMTzlSdriVZzH81Xwj3QEUxCp6

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDxTCCAq2gAwIBAgIBADANBgkqhkiG9w0BAQsFADCBgzELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpnbmExEzARBgNVBAcTClnjb3R0c2RhbGUxGjAYBgNVBAoT
EUdvRGFkZHkuY29tLCBjbMUMTEwLwYDVQQDEyhHbyBEYWRkeSBsb290IENlcnRp
ZmljYXRlIEF1dGhvcml0eSAAtIEcyMB4XDTE5MDkxMDAwMDAwMFoXDTM3MTIzMTIz
NTk1OVowgYmXcZAJBgNVBAYTAIVTMRAwDgYDVQQIEwdBcm16b25hMRMwEQYDVQQH
EwpTY290dHNkYWxlMR0wGAYDVQQKEXFHb0RhZGR5LmNvbSwgSW5jLjExMC8GA1UE

AxMoR28gRGfKZHkgUm9vdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkgLSBHMjCCASlw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL9xYgjx+lk09xvJGKP3gEly6SKD
E6bFIEMBO4Tx5oVJnyfq9oQbTqC023CYxzIBsQU+B07u9PpPL1kwIuerGVZr4oAH
/PMWdYA5UXvl+TW2dE6pjYIT5LY/qQOD+qK+ihVqf94Lw7YZFAXK6sOoBJQ7Rnwy
DfMAZiLiJWltNowRGLfTshxgtDj6AozO091GB94KPutdfMh8+7ArU6SSYmlRJQVh
GkSbJcypQ5Yj36w6gZoOKcUcqeldHraenjAKOc7xiID7S13MMuyFYkMlNAJWJwGR
tDtwKj9useiciAF9n9T521NtYJ2/LOdYq7hfRvzOxBsDPAnrSTFcaUaz4EcCAwEA
AaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYE
FDqahQcQZyi27/a9BUFuIMGU2g/eMA0GCSqGSIb3DQEBCwUAA4IBAQCZ21151fmX
WWcDYfF+OwYxdS2hII5PZYe096acvNjpL9DbWu7PdIxztDhC2gV7+AJ1uP2lsdeu
9tfeE8tTEH6KRtGX+rcuKxGrkLAngPnon1rpN5+r5N9ss4UXnT3ZJE95kTXWXwTr
giOrmgIttRD02JDHBHNA7XiloKmf7J6raBKZV8aPEjoJpL1E/QYVN8Gb5DKj7Tjo
2GTzLH4U/ALqn83/B2gX2yKQOC16jdFU8WnjXzPKej17CuPKf1855eJ1usV2GDPO
LPavTK33sefOT6jEm0pUBsV/fdUID+Ic/n4XuKxe9tQWskMJDE32p2u0mYRlynqI
4uJEvlz36hz1

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICVDCCAdugAwIBAgIQZ3SdjXfYO2rbIvT/WeK/zjAKBggqhkhjOPQQDAzBsMQsw
CQYDVQQGEwJHUjE3MDUGA1UECgwuSGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJlc2Vh
cmNoIEluc3RpdHV0aW9ucyBDQTEkMCIGA1UEAwwbSEFSSUNBIFRMUyBFQ0MgUm9v
dCBDQSAyMDIxMB4XDTIxMDIxOTExMDExMFoXDTQ1MDIxMzExMDEwOVowbDELMAkG
A1UEBhMCR1IxNzA1BgNVBAoMLkhlbGxlbmljIEFjYWRLbWljIGFuZCBSZXNIYXJj
aCBJbnN0aXR1dGlbnMgQ0ExJDAiBgNVBAMMG0hBUklDQSBUTFMgRUNDIFJvb3Qg
Q0EgMjAyMTB2MBAGByqGSM49AgEGBSuBBAAiA2IABDgl/rGgltJ6rK9JOtDA4MM7
KKrxcm11AEelhPyajmuqS7psBAqIXhfyVYf8MLA04jRYVxqEU+kw2anylnTDUR9Y
STHmME5gEYd103KUKe+bECUqqHgtvpBBWJAVcqeht6NCMEAwDwYDVR0TAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUyRtTgRL+BNuW0aq8mm+3oJUzbsowDgYDVR0PAQH/BAQD
AgGGMAoGCCqGSM49BAMDA2cAMGQCMBHervjcToiwqfAircJRQO9gcS3ujwLEXQNW
SaSS6sUUihCm0w2wqsosQJz76YJungIwK0eaB8bRwoF8yguWGEEbo/QwCZ61Iygn
nxS2PFOiTazPffpskcYqSUXm7LcT4Tps

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFpDCCA4ygAwIBAgIQOcqTHO9D88aOk8f0ZIk4fjANBgkqhkiG9w0BAQsFADBs
MQswCQYDVQQGEwJHUjE3MDUGA1UECgwuSGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
c2VhcmNoIEluc3RpdHV0aW9ucyBDQTEkMCIGA1UEAwwbSEFSSUNBIFRMUyBSU0Eg
Um9vdCBDQSAyMDIxMB4XDTIxMDIxOTExMDEwNTUzOFoXDTQ1MDIxMzExNTUzN1owbDEL
MAkGA1UEBhMCR1IxNzA1BgNVBAoMLkhlbGxlbmljIEFjYWRLbWljIGFuZCBSZXNI
YXJjaCBJbnN0aXR1dGlbnMgQ0ExJDAiBgNVBAMMG0hBUklDQSBUTFMgUINBIFJv
b3QgQ0EgMjAyMTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAIvC569l
mwVnlskNjLnQDmT8zulkGCyEf3dRywQRNrhe7Wlxp57kjqmXZ8FHws+RFjZiPTgE
4VGC/6zStGndLuwRo0Xua2s7TL+MjaQenRG56Tj5eg4MmOIjHdFOY9TnuEFE+2uv
a9of08WRiFukiZLRgeaMOVig1mlDqa2YUihu2wr7a89o+uOkXjpFc5gH6l8Cct4M
pbOfrrqkdx2z/IpZ525yZa31MJQjB/OCFks1mJxTuy/K5FrZx40d/JiZ+yykgmvw
Kh+OC19xXFyuQnspiYHLA6OZyoieC0AJQTPb5lh6/a6ZcMBaD9YThnEvdmn8kN3b

LW7R8pv1GmuebxWMevBLKKAiOIakbDakO/IwkfN4E8/BPzWr8R0RI7VDI4BkrcY
AuUR0YLbFQDMYTfBKnya4dC6s1BG7oKsnTH4+yPiAwBIcKMJJnkVU2DzOFytOOqB
AGMUuTNe3QvboEUHGjMJ+E20pwKmafTCWQWIZYVWrkvL4N48fS0ayOn7H6NhStYq
E613TBoYm5EPWNgGVMWX+Ko/IlqmhaZ39qb8HOLubpQzKoNqhArlT4b4UEV4AIHr
W2jjJo3Me1xR9BQsQL4aYB16cmEdH2MtiKrOokWQCPxrvrNQKlr9qEgYRtaQQJKQ
CoReaDH46+0N0x3GfZkYVvYnZS6NRcUk7M7jAgMBAAGjQjBAMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEFAPiI6ZgpJIKM+qTW8VX6iVNvRLuMA4GA1UdDwEB/wQE
AwIBhjANBgkqhkiG9w0BAQsFAAOCAGeAPpBIqm5iFSVmewzVjIuJndftTgfvnNAU
X15QvWiWkKQUEapobQk1OUAJ2vQJLDSle1mESSmXdMgHHkdt8s4cUCbjnj1AUz/3
f5Z2EMVGpdAgS1D0NTsY9FVqQRtHBmg8uwkiYtlfVUKqrFOFrJVWNlar5AWMxaja
H6NpvVMPxP/cyuN+8kyIhkdgGvMA9YCRotxDQpSbIPDRzbLrLFPCU3hKTWsuQZqP
JzLB5UkZv/HywouoCjxxKLR9YjYsTewfM7Z+d21+UPCfDtcRj88YxeMn/ibvBZ3P
zzfF0HvaO7AWhAw6k9a+F9sPPg4ZeAnHqQJyIkv3N3a6dcSFA1pj1bF1BcK5vZSt
jBWZp5N99sXzqnTPBIWUmAD04vnKJGW/4GKvyMX6ssmeVkjaf2WdhW+o45WxLM0
/L5H9MG0qPzVMiho7suuyWPEdr6sOBjhXlzPrjoiUevRi7PzKzMHVIf6tLITe7pT
BGIBnfHAT+7hOtSLIBD6Alfm78ELt5BGnBkpiNxvoEppaZS3JGWg/6w/zgH7IS79
aPib8qXPMThcFarmlwDB31qlpzmq6YR/PFGGoOtmUW4y/Twhx5duoXNTSpv4Ao8YW
xw/ogM4cKGR0GQjTQuPOAF1/sdwTsOEFy9EgqoZ0njnnkf3/W9b3raYvAwtt41dU
63ZTGI0RmLo=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICwzCCAqkqAwIBAgIBADAKBggqhkiOPQQDAjCBqjELMAkGA1UEBhMCR1IxDzAN
BgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
c2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXR0b3JpdHkxRDBCBgNVBAMTO0hl
bGxlbmljIEFjYWRLbWljIGFuZCBSZXNIYXJjaCBJbnN0aXR1dGlvbnMgRUNDIFJv
b3RDQSAyMDE1MB4XDTE1MDcwNzEwMzcwMloXDTQwMDYzMDEwMzcwMlowgaOxCzAJ
BgNVBAYTAkdSMQ8wDQYDVQQHEwZBdGhlbnMxRDBCBgNVBAoTO0hlbGxlbmljIEFj
YWRLbWljIGFuZCBSZXNIYXJjaCBJbnN0aXR1dGlvbnMgQ2VydC4gQXV0aG9yaXR5
MUQwQgYDVQQDEztlZWxsZW5pYyBBY2FkZW1pYyBhbmQgUmVzZW50aG9yaXR5
dXRpb25leVDQyBSb290Q0EgMjAxNTB2MBAGByqGSM49AgEGBSuBBAAiA2IABJKg
QehLgoRc4vgxEZmGZE4JJS+dQS8KjrVPdJWyUWRrjWvmP3CV8AVER6ZyOFB2lQJa
jq4onvktTpnvLEhvTCUp6NFxW98dwXU3tNf6e3pCnGoKVlp8aQuqgAkkbH7BRqNC
MEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFLLQj
C4KZJAEOnLvkdV2/+5cgk5kqMAoGCCqGSM49BAMCA2cAMGQCMGfOFmI4oqxiRaep
lSTAGiecmjvAwNW6qef4BENThe5SId6d9SWDPp5YSy/XZxMOIQIwBeF1Ad5o7Sof
TUwJCA3sS61kFyjndc5FZXIHf8siQQ6ME5g4mlRtm8rifOoCWCKR

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGCzCCA/OgAwIBAgIBADANBgkqhkiG9w0BAQsFAADCBpjELMAkGA1UEBhMCR1IxDzAN
BgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
IFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXR0b3JpdHkxQDA+BgNVBAMT
N0hlbGxlbmljIEFjYWRLbWljIGFuZCBSZXNIYXJjaCBJbnN0aXR1dGlvbnMgUm9v
dENBIDIwMTUwHhcNMTUwNzA3MTAxMTIxWhcNNDANjMwMTAxMTIxWjCBpjELMAkG
A1UEBhMCR1IxDzANBgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNh

ZGVtaWMgYW5kIFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXRob3JpdHkx
QDA+BgNVBAMTN0hlbGxlbmljIEFjYWwRbWljIGFuZCZBSZSNiYXJjaCBJbnN0aXR1
dGlvbnMgUm9vdENBIDlwMTUwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoIC
AQDC+Kk/G4n8PDwEXT2QNrCRONk8ZlrVbTkBSRq0t89/TSNTt5AA4xMqKKYx8ZEA
4yjsriFBzh/a/X0SWwGDD7mwX5nh8hKDgE0GPt+sr+ehiGsxr/CL0BgzuNtFajT0
AoAkKAoCFZVedioNmToUW/bLy1O8E00BiDeUJRtCvCLYjqOWXjrZMts+6PAQZe10
4S+nfK8nNLspfZu2zwnI5dMK/IhlZXQK3HMcXM1AsRzUtoSMTFDPaI6oWa7CJ06C
ojXdFPQf/7J31Ycvqm59Jcfnxssm5uX+Zwdj2EUN3TpZZTlYepKZcj2chF6IIBjV
9Cz82XBST3i4vTwri5WY9bPRaM8gFH5MXF/ni+X1NYEZN9cRCLdmvtNKzoNXADrD
gfgXy5I2XdGj2HUb4Ysn6npIQf1FGQatJ5lOwXBH3bWfgVMS5bGMSF0xQxfjjMZ6
Y5ZLKTBOhE5iGV48zpeQpX8B653g+IuJ3SWYPZK2fu/Z8VFRfS0myGlZYeCsargq
NhEEelC9MoS+L9xy1dcdFkfkR2YgP/SWxa+OAXqlD3pk9Q0Yh9muiNX6hME6wGko
LfiNaFGq46V3xqSQDqE3izEjR8EJC0tu93ib14L8hCCZSRm2Ekax+0VVFqmjZayc
Bw/qa9wfLgZy7IalEuQt218FL+TwA9MmM+eAws1CoRc0CwIDAQABo0IwQDAPBgNV
HRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUcRVnyMjJvXVd
ctA4GGqd83EkVAswDQYJKoZIhvcNAQELBQADggIBAHW7bVRLqhBYRjTyYtcWNlOI
XtVsyle9tC5G8jH4fOpCtZMWVdyhDBK2mF+D1hYc2Ryx+hFjtyp8iY/xnmMsVMI
M4GwVhO+5lFc2JsKT0ucVIMC6U/2DWDqTUVJ6HwbISHTGzrMd/K4kPFox/la/vot
9L/J9UUbzbjgQKjeKoa004wlshYaT/4mWJ3iBj2fjRnRUjtkNaeJK9E10A/+yd+2V
Z5fkscWrv2oj6NSU4kQoYsRL4vDY4ilrGnB+JGGTe08DMiUNRSQrlrRGar9KC/ea
j8GsGsVn82800vpzY4zvFrCopEYq+OsS7HK07/grfoxSwIuEVpKvPuNVqNxmsdnh
X9izjFk0WaSrT2y7HxjbdavYy5LNIIDhhDgcGH0tGEPEVvo2FXDtKK4F5D7Rpn0lQ
l033DIzdwJVqwbDG2jJ9SrcR5q+ss7FJej6A7na+RZukYT1HCjI/CbM1xyQVqdf
bzoEvM14iQuODy+jqk+iGxI9FghAD/FGTNeqewjBCvVtJ94Cj8rDtSvK6evIIVM4
pcw72Hc3MKJP2W/R8kCtQXoXxdZKNYm3QdV8hn9VTYNKpXMgwDqvKPGaIJ7ZjnHK
e7iG2rKPMt4dEw0SEe7Uq/DpFXYC5ODfqiAeW2GFZECpkJcNrVPSWh2HagCXZWK0
vm9qp/UsQu0yrbYhnr68

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIIFajCCA1KgAwIBAgIQld2szmKXlKFD6LDNDmpeYDANBgkqhkiG9w0BAQsFADBP
MQswCQYDVQQGEwJUVzEjMCEGA1UECgwaQ2h1bmdod2EgVGVSZWNvbSBDbY4sIEx0
ZC4xGzAzBgNVBAMMEkhpUEtJIFJvb3QgQ0EgLSBHMTAeFw0xOTAyMjIwMDRa
Fw0zNzEyMEZExNTU5NTlaME8xGzAjbG9uY2VhYXJjaCBJbnN0aXR1dGlvbnMgUm9vdENBIDlwMTUwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoIC
YSBUZwXlY29tIENvLiwgTHRkLjEbmBkGA1UEAwwSSGlQS0kgUm9vdCBDQSAIEcx
MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICGKCAgEA9B5/UnMyDHPkvRN0o9Qw
qNCuS9i233VHZvR85zkEHmpwINJaR3JnVfSl6J3VHiGh8Ge6zCFovkRTv4354twv
Vcg3Px+kwJyz5HdcoEb+d/oaoDjq7Zpy3iu9lFc6uux55199QmQ5eiY29yTw1S+6
lZgRZq2XNdZ1AYDgr/SEYYwNHI98h5ZeQa/rh+r4XfEuiAU+TCK72h8q3VJGZDnz
Qs7ZngyzsHeXZJzA9KMuH5UHsBffMNsAGJZMoYFL3QRtU6M9/Aes1MU3guklQgZ
KILSQjqj2FPseYlSGDIcpJQ3AOPgz+yQlda22rpEZfdhSi8MEyr48KxRURHH+CK
FgeW0iEPU8DtqX7UTuybCeyvQqww1r/REEXgphaypcXTT3OUM3ECoWqj1jOXTyFj
HluP2cFeRXF3D4FdXyGarYPM+l7WjSNfGz1BryB1ZlpK9p/7qxj3cc2HTHsOyDr
y+K49a6SsvfhEvyovKTmiKe0xRvNlS9H15ZFblzqMF8b3ti6RZsR1pl8w4Rm0bZ
/W3c1pzAtH2lsN0/Vm+h+fbkEkj9Bn8SV7apI09bA8PgcSojt/ewsTu8mL3WmKgM
a/aOEmem8rJY5AIJEzypuxC00jBF8ez3ABHfZfjcK0NVvxaXxA/VLGGEqnKG/uY6
fsI/fe78LxQ+5oXdUG+3Se0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNV

HQ4EFgQU8ncX+l6o/vY9cdVouSlGDDjYr7AwDgYDVR0PAQH/BAQDAgGGMA0GCSqG
SIlb3DQEBcWUAA4ICAQBQUfB13HAE4/+qddRxosuej6ip0691x1TPOhwEmSKsxBHi
7zNKpiMdDg1H2DfHb680f0+BazVP6XKlMeJ45/dOlBhbQH3PayFUhuaVevvGyuqc
SE5XCV0vrPStJczWNWseanMX/mF+lLFjfiRFOs6DRfQUJ748JzjkZ4Bjgs6Fza
ZsT0pPBWGTMPwMWSBUdGSquEwx4noR8RkpknDZMPvDY711ePJlsMu5wP1G4wB9Tc
XzZoZjmDlicmisjEO6faIW/Vcobpf2LlI07QJNBAsNB1CI69aO4I1258EHBG3zg
iLKecoaZAeO/n0kZtCW+VmWuF2PIHt/o/0elv+EmBYTksMCv5wiZqAxeJoBF1Pho
L5aPruJKHJwWDBNvOIf2u8g0X5IDUXlwpt/L9ZlNec1OvFefQ05rLisY+GpzjLrF
Ne85akEez3GoorKGB1s6yeHvP2UEgEcyRHCVTjFnanRbEEV16rCf0OY1/k6fi8wr
kkVbbiVghUbN0aqwdmaTd5a+g744tiROJgvM7XpWGuDpWsZkrUx6AEhEL7lAUXM+
vhV4nYWBSipX3tUZQ9rbyltHhoMLP7YNdnhzeSjesYAfz77RP1YQmCuVh6EfnWQU
YDksswBVLuT1sw5XxJFBAJw/6KXf6vb/yPCtbVKoF6ubYfwSUTXkJf2vqmqGOQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFzzCCA7egAwIBAgIUCBZfikyl7ADJk0DfxMaul7gcWqQwDQYJKoZIhvcNAQEL
BQAwbzELMAkGA1UEBhMCSEsxEjAQBGNVBAgTCUhvbmcgS29uZzESMBAGA1UEBxMJ
SG9uZyBlb25nMRYwFAyDVQQKEw1lb25na29uZyBQb3N0MSAwHgYDVQQDExdlb25n
a29uZyBQb3N0IFJvb3QgQ0EgMzAeFw0xNzA2MDMwMjI5NDZaFw00MjA2MDMwMjI5
NDZaMG8xCzAJBgNVBAYTAkhLMRIwEAYDVQQIEw1lb25nIetvbmcxEjAQBGNVBACt
CUhvbmcgS29uZzEWMBQGA1UEChMNSG9uZ2tvmcGU9ZzdDEgMB4GA1UEAxMXSG9u
Z2tvmcGU9ZzdCBSb290IENBIDMwggLiMA0GCSqGSIb3DQEBQUAA4ICDwAwggIK
AoICAQCziNfzqz8gTr7m1gNt7ln8wlffKWihgw4+aMdoWJwcYEuJQwy51BWY7sFO
dem1p+/l6TWZ5Mwc50tfjTMwIDNT2aa71T4Tjukfh0mtUC1Qyhi+AViiE3CWu4mI
VoBc+L0sPOFMV4i707mV78vH9toxdCim5lSj9UExyUUmGs2C4HDAOym71QP1mbpV
9WTRYA6ziUm4ii8F0oRFKHyPaFASepWltVPLwpgchKOesL4jpNrcyCse2m5FHomY
2vkALgbpDDtw1VAliJnLzXNg99X/NWfFobxeq81KuEXryGgeDQ0URhLj0mRiikKY
vLTGCAj4/ahMZJx2Ab0vqWwzD9g/KLg8aQFChn5pwckGyuV6RmXpwtZQQS4/t+Tt
bNe/JgERohYpSms0BpDsE9K2+2p20jzt8NYt3eEV7KObLyzJPivkaTv/ciWxNoZb
x39ri1UbSsUgYT2uy1DhCDq+sI9jQVMwCFk8mB13umOResoQUGC/8Ne8lYePl8X+
l2oBlKN8W4UdKjk60FSh0Tlxf0h+bV78OLgAo9uliQLKAeLKjEiafv7ZkGL7YK
TE/bosw3Gq9HhS2KX8Q0NEwA/RiTzxPRN+ZlItSgxVd7GYYKecsAyVKvQv83j+Gj
Hno9UKtjBucVtT+2RteUN7F+8kjDf8V1/peNRY8apxpyKBpADwIDAQABO2MwYTAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAFBgNVHSMEGDAWgBQXnc0e
i9Y5K3DTXNSguB+wAPzFYTAdBgNVHQ4EFgQUF53NHovWOSTw01zUoLgfsAD8xWEw
DQYJKoZIhvcNAQELBQADggIBAfbVe27mIghSQpsY1Q7XZiNc4/6gx5LS6ZStS6LG
7BJ8dNVI0lkUmcDrudHr9EgwW62nV3OZqdPlt9EuWSRY3GguLmLYauRwCy0gUCck
MpXRAJi70/33MvJJrsZ64Ee+bs7Lo3l6LWldy8joRTnU+kLBEUx3XZL7av9YROXR
gZ6voJmtvqkBZss4HTzfQx/0TW60uhdG/H39h4F5ag0zD/ov+BS5gLNdTaqX4fnk
GMX41TiMJz98iji7lpJiCzfeT2OnpA8vUFKOt1b9pq0zj8lMH8yfaIDlNDceqFS
3m6TjRgm/VWsvY+b0s+v54Ysyx8Jb6NvqYTUC79NoXQbTiNg8swOqn+knEwlqLJm
Ozj/2ZQw9nKEvmhVEA/GcywWaZMH/rFF7buiVWqw2rVKAiUnhde3t4ZEfolsGcs+
l6mc1X5VTMbeRRAC6uk7nwNT7u56AQIWeNTowr5GdogTPyK7SBIdUGC0An4hGh6c
JfTzPV4e0hz5sy229zdcxsshTrD3mUcYhcErulWuBurQB7Lcq9CClnXO0lD+mefP
L5/ndtFhKvshuzHQqp9HpLiYhY6UFfEW0NnxWViA0kB60PZ2Pierc+xYw5F9KBA
LJstxabArahH9CdMOA0uG0k7UvToiImrVCjU8jVStDKDYmlkJGcn5fqdBb9HxEG
mpv0

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFazCCA1OgAwIBAgIRAIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxFY2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBY
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBY
MTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVyg
h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPjxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgjU+g4rk8Kb4Mu0UlxjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBILDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/IJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
OlFuhjuefXKKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEvzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHwBfEbwrw
qHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkGJyKIrGF1XIpu+ILlaS/V9lZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK
NFtY2PwByVS5uCbMioziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIIInNzktHCgKQ5
ORAZI4JMPJ+GslWYHb4phowim57iazT0XoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/sKAiuvtd7u+Nxe5AW0wdeRlN8NwdC
jNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMlj4Ui0/1lvh+wjChP4kqKOJ2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SEeZwxA57d
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iltreGCc=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICGzCCAAGgAwIBAgIQQdKd0XLq7qeAwSxs6S+HUjAKBggqhkJOPQQDAzBPMQsw
CQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2gg
R3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00
MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAe
Fw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUz
EpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMT
DElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjB
PMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2
ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA
5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQG
EwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATB
gNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwND
M4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZ
WYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw
00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw
0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEp
MCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDE
lUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQ
swCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3
JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTc
xNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdX
JpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQw
MDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UECh
MgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9v
dCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQ
QGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFT
ATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MT
EwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgU
mVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMD
BaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50
ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUj
AeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJV
UzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVB
AMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4W
jBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy
2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MD
A5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQG
EwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATB
gNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwND
M4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZ
WYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw
00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw
0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEp
MCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDE
lUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQ
swCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3
JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTc
xNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdX
JpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQw
MDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UECh
MgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9v
dCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQ
QGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFT
ATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MT
EwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgU
mVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMD
BaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50
ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUj
AeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJV
UzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVB
AMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4W
jBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy
2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MD
A5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQG
EwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATB
gNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwND
M4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZ
WYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw
00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw
0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEp
MCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDE
lUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQ
swCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3
JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTc
xNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdX
JpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQw
MDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UECh
MgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9v
dCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQ
QGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFT
ATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MT
EwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgU
mVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMD
BaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50
ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUj
AeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJV
UzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVB
AMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4W
jBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy
2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MD
A5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQG
EwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATB
gNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwND
M4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZ
WYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw
00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw
0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEp
MCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDE
lUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQ
swCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3
JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTc
xNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdX
JpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQw
MDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UECh
MgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9v
dCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQ
QGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFT
ATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MT
EwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgU
mVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMD
BaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50
ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUj
AeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJV
UzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVB
AMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4W
jBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy
2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MD
A5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQG
EwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATB
gNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwND
M4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZ
WYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw
00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw
0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEp
MCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDE
lUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQ
swCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3
JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTc
xNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdX
JpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQw
MDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UECh
MgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9v
dCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQ
QGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFT
ATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MT
EwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgU
mVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMD
BaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50
ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUj
AeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJV
UzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVB
AMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4W
jBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy
2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MD
A5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU
2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA
5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA
1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcg
Um9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQ
YDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3Jvd
XAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMDAwMDBaFw00MDA5MTcxNj
A0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJp
dHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdCBYUjAeFw0yMDA5MDQwMD
AwMDBaFw00MDA5MTcxNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMg
SW50ZXJuZXQgU2VjdXJpdHkgUmVzZWYy2ggR3JvdXAFTATBgNVBAMTDElUkcgUm9vdC
BYUjAe

3j6HEDbhuaR672BQssvKplbgN6+rNBM5Jeg5ZuSYeqoSmJxZZoY+rfGwyj4GD3vw
EUs3oERte8uojHH01bWRNsZWfCYr3lEXsZdMUD2xlv18BX0tldUAvwFno157plzy
9yLxkA2T26pEUWbMfXYD62qoKjgZl3YNa4ph+bz27nb9cCvdKTz4Ch5bQhyLVi9V
GxyhLrXHFub4qjySjmm2AcG1hp2JDws4lFTo6tyePSW8Uybt1as5qsVATFSrsrTZ
2fjXctscvG29ZV/viDUqZi/u9rNl8DONfJhBaUYPQxxp+pu10GFqzcpL2UyQRqsV
WaFHVCKugyhFHMkiQ3IXAAaOReyL4jM9f9oZROricsPflsbyVtTdX5Vy7W1f90gD
W/3FKqD2cyOEEBsB5wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQU43HgntinQtnbcZFrlJPrw6PRFKMwDQYJKoZIhvcN
AQELBQADggIBAef63QqwEZE4rU1d9+UOl1QZgkiHVlyqZJnYWv6IAcVYpZmxI1Qj
t2odIFlAWJBF9MJ23XLbLSQdf4an4EKwt3X9wnQW3IV5B4Jaj0z8yGa5hV+rVHV
DRDtULAJ+7AmgjVQDZcDiFpboBhDhXAuM/FSRJSzL46zNQuOAXeNf0fb7iAaJg9
TaDKQGXS3z1i9kKIT/YPyNtGtEqJBnZhbMX73huqVjRI9PHE+1yJX9dsXNw0H8G
lwmEKYBhHfpe/3OsoOOJuBxxFcbMX8S3Ofm6/n6J91eEyrRjuazr8FGF1NFTwW
mhlQBjQym9li1JfPFgEKCAZmExfrngdbkaqIHWchezxQMxNRF4eKLg6TCMf4Df
WN88uieW4oA0beOY02QnrEh+KHdcxiVhJfiFDGX6xDIvpZgF5PgLZxYWxoK4Mhn5
+bl53B/N66+rDt0b20XkeuC4pVd/GnwU2lhlXV5C15V5jgclKlZM571cXR5f1GJ
tshquDDIajjDbp7hNxbqBWJMWxJH7ae0s1hWx0nzfxJoCTFx8G34Tkf71oXuxVhA
GaQdp/LQzfcFpPz+vCZHTetBXZ9FRUGi8c15dxVJCO2SCdUyt/q4/i6jC8UDfv
8Ue1fXwsBOxonbRJRBD0ckscZOf85muQ3Wl9af0AVqW3rLatt8o+Ae+c
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF8TCCA9mgAwIBAgIQALC3WhZIX7/hy/WL1xnmfTANBgkqhkiG9w0BAQsFADA4
MQswCQYDVQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCKl6
ZW5wZS5jb20wHhcNMDCxMjEzMTMwODI0WWhcNMZcxMjEzMDgyNzI1WjA4MQswCQYD
VQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCKl6ZW5wZS5j
b20wgglMA0GCSqSISIbDQEBQUAA4ICDwAwggIKAoICAQDJ03rKDx6sp4boFmVq
scIbRTJxldn+EFvMr+eleQGPicPK8lVx93e+d5TzcqQsRNiekpsUOqHnJJAKClAO
xdgmlOHZSOEtPtoKct2jmRXagaKH9HtuJneJWK3W6wyyQXpzbm3benhB6QiEn6H
LmYRY2xU+zydcsC8Lv/Ct90NduM61/e0aL6i9eOBbsFGb12N4E3GVFWJGjMxCrFX
uaOKmMPsOzTFIUFpfnXCPCDFYbpRR6AgkJOhkEvzTnyFRVSA0QUmQbC1TR0zvsQD
yCV8wXDbO/QJLVQnSKwv4cSsPsjLkxTOTcj7NMB+eAJRE1NZMDhDVqHlrytG6P+
JrUV86f8hBnp7KGItERphIPzidF0BqnMC9bC3ieFUCbKF7jJeodWLBoBHmy+E60Q
rLUk9TiRodZL2vG70t5HfG8gfZZa88ZU+mNFctKy6lvROUbQc/hhqqK0GqfvEyN
BjNaooXlkDWgYlWWTvDjovoDGrQscbNYLN57C9saD+veIR8GdwYDsMnvmfzAuU8L
hij+0rnq49qlw0dpEuDb8PYzi+17cNcC1u2HGCgsBCRMd+RlihrGO5rUD8r6ddIB
QFqNeb+Lz0vPqhbBleStTio+F5HUsWLlguWABKQDfo2/2n+iD5dPDNMN+9fR5XJ+
HMh3/1uaD7euBUbl8agW7EekFwIDAQABo4H2MIHzMIGwBgNVHREEgagwgaWBD2lu
Zm9AaXplbnBILmNvbaSBkTCBjjFHMEUGA1UECgw+SVpFTlBFIFMuQS4gLSBDSUYg
QTAXMzM3MjYwLVJNZXJjLlZpdG9yaWEtR2FzdGVpeiBUMTA1NSBGNjIjIzZG9yYmlkZWVMTQgLSAUMTAx
BgNVBAkMOKF2ZGEgZGVsIE1lZGl0ZXJyYW5lbyBFdG9yYmlkZWVMTQgLSAUMTAx
MCBWAxRvcmlhLUdhc3RlaXowDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVROBBYEFB0cZQ6o8iV7tJHP5LGx5r1VdGwFMA0GCSqSISIbDQEBQUAA4ICAQB4pgwWSp9MiDrAyw6lFn2fuUhfGI8NYjb2zRlrrKvV9pF9rnHzP7MOeIWb
laQnIUdCSnxIOvVFfLMMjlf4rJUT3sb9fbgkEyrkgPH7UIBzg/YsfqikuFgba56
awmqxinuaElnMIAkejEWOVt+8Rwu3WwJrflxwYJOubv5vr8qhT/AQKM6WfxZSzw
JNu0FXWuDYi6LnPAvViH5ULy617uHjAimcs30cQhbIHsvm0m5hzkQiCeR7Csg1lw

LDXWrzY0tM07+DKo7+N4ifuNRSzanLh+QBxh5z6ikixL8s36mLYp//Pye6kfLqCT
VyvehQP5aTfLnnhqBbTFMXij7HqnheG5ezzevh55hM6fcA5ZwjUukCox2eRFekGk
LhObNA5me0mrZJfQRsN5nXJQY6aYWwa9SG3YOYNw6DXwBdGqvOPbyALqfP2C2sJb
UjWumDqtujWTI6cfSN01RpiyEGjkpTHCClguGYEQyVB1/OpaFs4R1+7vUIgtYf8/
QnMFIEPVjjxOAToZpR9GTnfQXeWBlIGH/pR9hNiTrdZoQ0iy2+tzJOeRf1SktoA+
naM8THLCV8Sg1Mw4J87VBp6iSNnpn86CcDaTmjvfliHjWbcm2pE38P1ZWroZYGls
QyYBNWNgVYkDOnXYukrZVP/u3oDYLdE41V4tC5h9Pmzb/CaIxx==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIECjCCAvKgAwIBAgIJAMJ+QwRORz8ZMA0GCSqGSIb3DQEBCwUAMIGCMQswCQYD
VQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3QxZjAUBGNVBAoMDU1pY3Jvc2VjIEExO
ZC4xZjZlbnBAMMHk1pY3Jvc2VjIGUtU3ppZ25vIFJvb3QgQ0EgMjAwOTefMB0G
CSqGSIb3DQEJARYQaW5mb0BILXN6aWduby5odTAeFw0wOTA2MTYxMTMwMTThaFw0y
OTEyMzAxMTMwMTThaMIGCMQswCQYDVQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3Qx
ZjAUBGNVBAoMDU1pY3Jvc2VjIEExOZC4xZjZlbnBAMMHk1pY3Jvc2VjIGUtU3pp
Z25vIFJvb3QgQ0EgMjAwOTefMB0GCSqGSIb3DQEJARYQaW5mb0BILXN6aWduby5o
dTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOn4j/NjrdqG2KfgQvvP
kd6mJviZpWNwrZuuyjNAfW2WbqEORO7hE52UQIKavXWFdCyoDh2Tthi3jCyoZ/tc
cbna7P7of0/kLx2yqHWH2Leh5TvPmUpG0IMZfcChEhyVbUr02MeITTMuhTlAdX4U
fIASmFDHQWe4oIBhVKZsTh/gnQ4H6cm6M+f+wFUoLAKApxn1ntxVUwOXewdl/5n7
N4okxFnMUBBjjqqpGrCEGob5X7uxUG6k0QrM1XF+H6cbfPVTbiJfyyvm1HxdrbC
xkzIBQHZ7Vf8wSN5/PrIJOV87VqUQHqD9bpEqH5GoP7ghu5sJf0dgYzQ0mg/wu1
+rUCAwEAAaOBgDB+MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0G
A1UdDgQWBBTLDB8fQkPMPcu1SCOhGnqmKrs0aDAfBgNVHSMEGDAWgBTLDB8fQkPM
Pcu1SCOhGnqmKrs0aDABgNVHREEFDASgRBpbmZvQGUTC3ppZ25vLmh1MA0GCSqG
SIb3DQEBCwUAA4IBAQDJ0Q5eLtXMs3w+y/w9/w0lZMEyL/azXm4Q5DwpL7v8u8h
mLzU1F0G9u5C7DBsoKppyvGvivo/C3NqPuouQH4frlRheesuCDfXI/OMn74dseGk
ddug4lQUsbocKaQY9hK6ohQU4zE1yED/t+AFdlfBHFny+L/k7SViXITwfn4fs775
tyERzAMBvNcNEJIEGzSBHq2cGsMEPO0CYdYeBvNfOofyK/FFh+U9rNHHV4S9a67c
2Pm2G2JwC0z2yULyMtd6YebS2z3PyKnJm9zbWETXbzivf3jTo60adbocwTZ8jx5t
HMN1Rq41Bab2XD0h7lbwyYiLXpUq3DDfSJlgnCW

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICWTCCAd+gAwIBAgIQZvI9r4fei7FK6gxXMqHC7DAKBggqhkJOPQQDAzBlMQsw
CQYDVQQGEwJlUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcnF0aW9uMTYwNAYD
VQQDEy1NaW5yb3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDlw
MTcwHhcNMtkxMjE4MjMwNjQ1WhcNNDIwNzE4MjMxNjA0WjBlMQswCQYDVQQGEwJlUz
EeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcnF0aW9uMTYwNAYDVQQDEy1NaW5y
b3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDlwMTcwZjAQBgcq
hkjOPQIBBgUrgQQAIGNiAATUvD0CQnVBEyPNgASGAIEvaqiBYgtlzPbKnR5vSmZR
ogPZnZH6thaxjG7efM3beaYvzrvOcS/lpaso7GMEZpn4+vKTEAXhgShC48Zo9OYb
hGBKia/teQ87zvH2RPUBeMCjVDBSMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UdDgQWBBTlY5lycFIM+Oa+sgRXKSrPQhDtNTAQBgkrBgEEAYI3
FQEEAwIBADAKBggqhkJOPQQDAwNoADBIAjBY8k3qDPlfXu5gKcs68tvWMOQZP3zV

L8KxzJOuULsJMsbG7X7JNpQS5GiFBqIb0C8CMQCZ6Ra0DvpWSNSkMBaReNtUjGUB

iudQZsIxtzm6uBoiB078a1QWIP8rtedMDE2mT3M=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFqDCCA5CgAwIBAgIQHtOXCv/YtLNHcB6qvn9FszANBgkqhkiG9w0BAQwFADBl
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmlF0aW9uMTYw
NAYDVQQDEy1NaWNyb3NvZnQgUINBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5
IDIwMTcwHhcNMTkxMjE4MTIyWhcNNDIwNzE4MjMwMDIzWjBlMQswCQYDVQQG
EwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmlF0aW9uMTYwNAYDVQQDEy1N
aWNyb3NvZnQgUINBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIwMTcwggli
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDKW76UM4wplZEWCPw9R2LBifOZ
Nt9GkMml7Xhqb0eRaPgnZ1AzHaGm++DIQ6OEAlcBXZxIQIJTEly/xztokLaCLeX0
ZdDMbRnMlfl7rEqUrQ7eS0MdhweSE5CAG2Q1OQT85elss7YfUJQ4ZVBcF0a5toW1
HLUX6NZFndiyJrDKxHBKrmCk3bPZ7Pw71VdyvD/IybLeS2v4I2wDwAW9lcfNcztm
gGTjGqwu+UcF8ga2m3P1eDNbx6H7JyqhtJqRjJHTOoI+dkC0zVJhUXAoP8XFWvLJ
jEm7FFtNyP9nTUwSlq31/niol4fX/V4ggNyhSyL71Imtus5Hl0dVe49FyGcohJUc
aDDv70ngNXtk55iwlNpNhTs+VcQor1fznhPbRiefHqJeRIOkpcrVE7NLP8TjwuaG
YaRSMlI6IE9vDzhTyzMMEyuP1pq9KsgtsRx9S1HKR9FIJ3Jdh+vVReZIZZ2vUpC6
W6IYZVcSn2i51BVrlMRpIjp0M+Dt+VGOQVDJNE92kKz8OMHY4Xu54+OU4UZpyw4K
UGsTuqwPN1q3ErWQgR5WrlcihtnJ0tHXUeOrO8ZV/R4O03QK0dqq6mm4lyiPSMQH
+FJDOvTKVTUssKZqwJz58oHhEmrARdlns87/I6KJCITUFLkqqNfs+avNJVgyeY+Q
W5g5xAgGwax/Dj0ApQIDAQABo1QwUjAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUcCtZf4ayci8awznjwNnpv7tNsiMwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEMBAQgIBAKyPl3CEZaJjqPnktaXFbgToqZC
LgLNfGvZJ8og6Lq46BrsTaiXVq5lQ7GPAJtSzVXNUzltYkyLDVt8LkS/gxCP81OC
gMNPOsduET/m4xaRhPtthH80dK2Jp86519efhGSSvpWhrQITM93uCupKUY5vVau6
tZRGrox/2KJQJWVvgEbbMwSubLWYdFQl3Jpk+ONVFT24bcMKpBLBaYVu32TxU5nh
SnUgnZUP5NbcA/FZGOHhIBJXWpS2qdgXKxdJ5XbLwVaZOjex/2kskZGT4d9Mozd2
TaGf+G0eHdP67Pv0RR0Tbc/3WeUiJ3IrhvNXuzDtJE3cfVa7o7P4NHmJweDyAmH3
pvwPuxwXC65B2Xy9J6P9LjrRk5Sxc0ki69bIImtt2dmefU6xqaWM/5TkshGsRGR
xpl/j8nWZjEgQRCHLQzWwa80mMpkg/sTV9HB8Dx6jKXB/ZUhoHHBk2dxEuqPiApp
GWSZI1b7rCoucL5mxAYe7+WL85MB+GqQk2dLsmijtWKP6T+MejteD+eMuMZ87zf9
dOLITzNy4ZQ5bb0Sr74MTnB8G2+NszKTc0QWbej09+CVgl+WXTik9KveCjChk9hN
AHFiRsdLokKEW39lt2c0Ui2cFmuqqNh7o0JMcccMyj6D5KbvtwEwXlGjefVwaaZB
RA+GsCyRxj3qrg+E

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFojCCA4qgAwIBAgIUAZQwHqIL3fXFMypqxQ0Rx+NZQTQ0wDQYJKoZIhvcNAQEM
BQAwA TELMAkGA1UEBhMCS1IkJAkBgNVBAoMHU5BVkVSIEJVU0IORVNTIFBMQVRG
T1JNIEVncnAuMTIwMAYDVQQDDClOQVZFUiBhBg9iYWwgUm9vdCBDZXJ0aWZpY2F0
aW9uIEF1dGhvcml0eTAeFw0xNzA4MTgwODU4NDJaFw0xNzA4MTgyMzU5NTlaMGkx
CzAJBgNVBAYTAktSMSYwJAYDVQQKDB1OQVZFUiBCVVNJTkVTUyBQTEFURk9STSBD
b3JwLjEyMDAGA1UEAwppTkFWRVlIgR2xvYmFsIFJvb3QgQ2VydGlmaWNhdGlvbiBB
dXR0b3JpdHkwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC21PGTXLVA

iQqrDZBbUGOukJR0F0Vy1ntIWilLp1agS7gvQnXp2XskWjFlqxcX0TM62RHcQDaH
38dq6SZeWYp34+hInDEW+j6RscrJo+KfziFTowI2MMtSAuXaMl3Dxeb57hHHi8IE
HoSTGEq0n+USZGnQJoViAbbJAh2+g1G7XNr4rRVqmfeSVpC0W+m/6imBEtRTkZaz
kVrd/pBzKPswRrXKCAfHcXLJZtM0l/aM9BhK4dA9WkW2aacp+yPOiNgSnABlqKYP
szuSjXE0dMWLyEz59JuOuDxp7W87UC9Y7cSw0BwbagzivESq2M0UXZR4Yb8Obtoq
vC8MC3GmsxY/nOb5zJ9TNeIDoKAYv7vxvTWjLcNQvcGufFt7QSUqP620wbGQGhf
nZ3zVHbOUzoBppJB7ASjjw2i1QnK1sua8e9DXcCrpUHPXFNwcMmIpi3Ua2FzUCaG
YQ5fG8lr4ozVu53BA0K6lNpfqbDKzE0K70dpAy8i+/Eozr9dUGWokG2zdLAIx6yo
0es+nPxdGoMuK8u180SdOqcXYZaicdNwlhVNt0xz7hlcxVs+Qf6sdWA7G2POAN3a
CJBitOUt7kinaxeZVL6HSuOpXgRM6xBtVNbv8ejyYhbLgGvtPe31HzClrkVJE+2K
AQHJuFFYwGY6sWZLxNUxAmLpdIQM201GLQIDAQABo0IwQDAdBgNVHQ4EFgQU0p+I
36HNL3s9TsBAZMzJ7LrYEswDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMB
Af8wDQYJKoZIhvcNAQEMBAQggIBADLkGLodPVQG3dLSLvCkASELZ0jKbY7gyKoN
qo0hV4/GPnrK21HUUrPUloSIWGB/5QuOH/XcChWB5Tu2tyIvCZwTFRFsDDUIbatj
cu3cvuzHV+YwIHHW1xDBE1UBjCpD5EHxzzp6U5LOogMFDTjfArsQLtk70pt6wKGm
+LUx5vR1yblTmXVHIloUFcd4G7ad6Qz4G3bxhYTEodoS76TiEJd6eN4MUZeoIUCL
hr0N8F5OSza7OyAfikJW4Qsav3vQIkMsRIz75Sq0bBwcupTgE34h5prCy8VCZLQe
lHsIJchxzIdFV4XTnyliIoNRlwAYl3dqmJLJfGBs32x9SuRwTMKeuB330DTHD8z7
p/8Dvq1wkNoL3chtl1+afwkyQf3NosxabUzyqkn+Zvjp2DXrDige7kgvOtB5CTh8
piKCK5XQA76+AqAF3SAi428diDRgxuYKuQl1C/AH6GmWNcf7I4GOODm4RStDeKLR
LBT/DSHycpWbXgnbiUSYqqFJu3FS8r/2/yehNq+4tneI3TqkbZs0kNwUXTC/t+sX
5Ie3cdCh13cV1ELX8vMxmV2b3RZtP+oGI/hGoiLtk/bdmuYqh7GYVPEi92tF4+KO
dh2ajcQGjTa3FPOdVgm3jjzVpG2Tgbet9r1ke8LJaDmgkpzNNIaRkPpkUZ3+/uul
9XXeifdy

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIEFTCCAav2gAwIBAgIGSUEs5AAQMA0GCSqGSIb3DQEBCwUAMIGnMQswCQYDVQQG
EwJIVITERMA8GA1UEBwwIQnVkYXBlc3QxFTATBgNVBAoMDE5ldExvY2sgS2Z0LjE3
MDUGA1UECwwuVGFu7pzw610dsOhbnlraWfkw7NrIChDZXJ0aWZpY2F0aW9uIFNl
cnZpY2VzKTE1MDMGA1UEAwwsTmV0TG9jayBBcmFueSAoQ2xhc3MgR29sZCkgRswR
dGFu7pzw610dsOhbnkwHhcNMDgxMjExMTUwODIxWhcNMjg1MjE1MTUwODIxWjCB
pzELMAkGA1UEBhMCSFUxETAPBgNVBACMECjE1ZGFwZXR0MRUwEwYDVQQKDAxOZXRM
b2NrIEtmdC4xNzA1BgNVBAsMLIRhbsO6c8OtdHbDoW55a2lhZMOzayAoQ2VydGlm
aWNhdGlvbIjBTZXJ2aWNLcykxNTAzBgNVBAMMLE5ldExvY2sgQXJhbnkgKENsYXNz
IEdvbGQpIEBfKXRhbsO6c8OtdHbDoW55MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAXCRec75LbRTDofTjl5Bu0jBFHjzuZ9lk4BqKf8owyoPjIMHj9DrT
lF8afFttvzBPhCf2nx9JvMaZCpDyD/V/Q4Q3Y1GLEqVw/HpYzY6b7cNGbIRwXdrz
AZAj/E4wqX7hJ2Pn7WQ8oLjJM2P+FPD/sLj916jAwJRDC7bVWaaeVtAkH3B5r9s5
VA1lddkVQZQBr17s9o3x/61k/iCa11zr/qYfCGSji3ZVrR47KGAuhyXoqq8fxmRG
ILdfwzZeSNuWU7c5d+Qa4scWhHaXWy+7GRWF+GmF9ZmnqfI0p6m2pgP8b4Y9VHx2
BJtr+UBdADTHLpl1neWIA6pN+APSQnbAGwIDAKiLo0UwQzASBgNVHRMBAf8ECDAG
AQH/AgEEMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUzPpnk/C2uNClwB7zU/2M
U9+D15YwDQYJKoZIhvcNAQELBQADggEBAKt/7hwWqZw8UQCgwBEIBaeZ5m8BiFRh
bvG5GK1Krf6BQCOUL/t1fC8oS2IkgyIL9WHxHG64YTjrgfpioTtaYtOUZcTh5m2C
+C8lcLlhJsFyUR+MLMOEkMNaj7rP9KdlpeuY0fsFskZ1FSNqb4VjMIDw1Z4fKRzC
bLBQWV2QWzuoDTDPv31/zvGdg73JRm4gpvlhUbohL3u+pRVjodSVh/GeufOJ8z2F

uLjbvrW5KfnaNwUASZQDhETnv0Mxz3WLJdH0pmT1kvarBes96aULNmLazAZfNou2

XjG4Kvte9nHfRCAexOYNkbQudZWAUWpLMKawYqGT8ZvYzsRjdT9ZR7E=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDtTCCAp2gAwIBAgIQdrEgUnTwhYdGs/gjGvbCwDANBgkqhkiG9w0BAQsFADBt
MQswCQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxmZT0lTVEUg
Rm91bmRhdGlvb1BFbmRvcnNlZDEoMCYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9i
YWwgUm9vdCBHQiBDQTAeFw0xNDEyMDExNTAwMzJaFw0zOTEyMDExNTEwMzFaMG0x
CzAJBgNVBAYTAkNIMRAwDgYDVQQKEwdXSXVNI2V5MSIwIAAYDVQQLExlPSVNURSBG
b3VuZGF0aW9uIEVuzG9yc2VhMSgwJgYDVQQDEx9PSVNURSBXSXVNI2V5IEdsb2Jh
bCBSb290IEEdCIENBMIIIBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2Be3
HEokKtaXscriHvt9OO+Y9bI5mE4nuBFde9lIlliCFSZqGzG7qFshISvYD06fWvGx
WuR51jIjK+FTzJlFXHtPrby/h0oLS5daqPZI7H17Dc0hBt+eFf1Biki3IPShehtX
1F1Q/7pn2COZH8g/497/b1t3sWtuuMlk9+HKQUYOKXHQuSP8yYFfTvdv37+ErXNk
u7dCjmn21HYdfp2nuFeKUWdy19SouJVUQHMD9ur06/4oQnc/nSMbsrY9gBQHtC5P
99UKFg29ZkM3fiNDecNAhvVMKdqOmQ0NpQSHiB6F4+IT1ZvliwNjeOvgGUpuuy9r
M2RYk61pv48b74JixwIDAQAB01EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUNQ/INmNe4qPs+TtmFc5RUuORmj0wEAYJKwYBBAGCNxUB
BAMCAQAwDQYJKoZIhvcNAQELBQADggEBAEBM+4eymYGGfp3FsLAmzYh7KzKNbrgh
cViXfa43FK8+5/ea4n32cZiZBKpDdHij40lhPnOMTzTg+XHEthYOU3gf1qKHLwI5
gSk8rxWYITD+KJAAjNHhy/peyP34EEY7onhCkRd0VQreUGdNZtGn//3ZwLWoo4rO
ZvUPQ82nK1d7Y0Zqqi5S2PTt4W2tKZB4SLrhI6qjiey1q5bAtEuiHZehevJuQHfHf
aPFITc58Bd9TZaml8LGBHAvRgOY1NK/VLSgWH1Sb9pWJmLU2NuJMW8c8CLC02Ic
Nc1MaRVUGpCY3useX8p3x8uOPUNpnJpY0CQ73xtAln41rYHHTnG6iBM=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICaTCCAe+gAwIBAgIQISpWdK7aDKtArb8roi066jAKBggqhkiG9w0BAQsFADBT
CQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxmZT0lTVEUgRm91
bmRhdGlvb1BFbmRvcnNlZDEoMCYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9iYWwg
Um9vdCBHQyBDQTAeFw0xNzA1MDkwOTQ4MzRaFw00MjA1MDkwOTU4MzNaMG0xCzAJ
BgNVBAYTAkNIMRAwDgYDVQQKEwdXSXVNI2V5MSIwIAAYDVQQLExlPSVNURSBGb3Vu
ZGF0aW9uIEVuzG9yc2VhMSgwJgYDVQQDEx9PSVNURSBXSXVNI2V5IEdsb2JhbCBS
b290IEEdDIENBMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAETOlQwMYPchi82PG6s4ni
eUqjFqdrVCTbUf/q9Akkwswsin8tqJ4KBDdLArzHkdIJuyiXZjHwD8dvQmqJLIX4W
p2OQ0jnUsYd4XxiWD1AbNTcPasbc2RNNpI6QN+a9WzGRo1QwUjAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4EFgQUSIcUrOPDnpBgOtfKie7T
rYy0UGYwEAYJKwYBBAGCNxUBBAMCAQAwCgYIKoZIzj0EAwMDaAAwZQIwJsdpW9zV
57LnyAyMjMPdeYwbY9XJUpROTYJKcx6ygISpJcBMWm1JKWB4E+J+SOTkAJEA2zQg
Mgj/mkkCtojeFK9dbJlxjRo/i9fgojaGHAeConZT/cKi7e97sIBPWA9LUzm9

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUeFhLq0sGUvjNwc1NBMotZbUZZMwDQYJKoZIhvcNAQEL

BQAwSDELMaKGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpbl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMSBHMzAeFw0xMjAxMTIxNzI3NDRaFw00
MjAxMTIxNzI3NDRaMEgxCzAJBgNVBAYTAkJNMRkwFwYDVQQKEsBRdW9WYWRRpcyBM
aW1pdGVkMR4wHA YDVQQDEsVRdW9WYWRRpcyBSb290IENBIDEGRzMwggLiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCgvlAQjunybEC0BJyFuTHK3C3kEakEPBtV
wedYMB0ktMPvhd6MLOHBPd+C5k+tR4ds7FtJwUrVu4/sh6x/gpqG7D0DmVIB0jWe
rNrWU8lmPNSsAgHaJNM7qAJGr6Qc4/hzWHa39g6QDbXwz8z6+cZM5cOGMAqNF341
68Xfuw6cwI2H44g4hWf6Pser4BOcBRiYz5P1sZK0/CPTz9XEJ0ngnjybCKOLXSoH
4Pw5qlPafX7PGglTvF0FBM+hSo+LdoINofjSxxR3W5A2B4GbPgb6U15jxaYA/qXp
UhtStZI5cgMJYr2wYBZupt0lwgNm3fME0UDiTouG9G/lg6AnhF4EwfWQvTA9xO+o
abw4m6SkltFi2mnAAZauy8RRNOoMqv8hJlmPSlzkYZqn0ukqeI1RPToV7qJzJqlc
3sX5kCLliEVx3ZGZbHqfPT2YfF72vhZooF6uCyP8Wg+qInYtyaEQHeTTRCOQIj/G
KubX9ZqzWB4vMikIG1SitZgj7Ah3HJVdYdHLiZxfokqRmu8hqkkWCKi9YSgxyXSt
hfbZxbGL0eUQMk1fiyA6PEkfM4VZDdvLCXVDA XP7a3F98N/ETH3Goy7IlXnLc6KO
Tk0k+17kBL5yG6YnLUlamXrXXAkgt3+UuU/xDRxeiElbEbfknduebPRq34wGmAot
zCjvpUfzUwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAWIB
BjAdBgNVHQ4EFgQUo5fW816iEOGrRZ88F2Q87gFwnMwwDQYJKoZIhvcNAQELBQAD
ggIBABj6W3X8PnrHX3fHyt/PX8MSxEBd1DKquGrX1RUVRpgjpeaQWxiZTOOtQqOC
MTalzen7xASWSIsBx40Bz1szBpZGZnQdT+3Btrm0DWHMY37XLneMlhwlI2hrhVd2
cDMT/uFPpiN3GPOajOi9ZcnPP/TJF9zrx7zABC4tRi9pZsMbj/7sPtPKIL92CiUN
qXsCHKnQO18LwIE6PWThv6ctTr1NxNgpxiIY0MWscgKCP6o6joilzHdCGPDdRS5
YCgtW2jgFqlmgiNR9etT2DGbe+m3nUvriBbP+V04ikkwj+3x6xn0dxoxGE1nVGww
b2X52z3sIexe9PSLymBIVNFxZPT5pqOBmZYZcfCkeF9OrYMh3jRjehZrJ3ydl02
8hP0r+AJx2EqbPfgna67hkooby7utHnNkDPDs3b69fBsnQGQ+P6Q9pxyz0fawx/k
NSBT8lTR32GDpgLiJTjehTItXnOQUl1CxM49S+H5GYQd1aJQzEH7QRTDvdbJWqNj
ZgKA vQU6O0ec7AAmTPWIUb+ol38YB7AL7YsmoWTTYUrrXJ/es69nA7Mf3W1daWhp
q1467HxpvMc7hU6eFbm0FU/DIXpY18ls6Wy58yljXrQs8C097Vpl4KlbQMJI mYFt
nh8GKjwStIsPm6lk8KaN1nrgS7ZklmOVhMJKzRwuJlczYOXD

-----END

CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIIFtZCCA5+gAwIBAgICBQkwDQYJKoZIhvcNAQEFBQAwRTELMAKGA1UEBhMCQk0x
GTAXBgNVBAoTEFF1b1ZhZGlzIExpbl0ZWQxGzAZBgNVBAMTElF1b1ZhZGlzIFJv
b3QgQ0EgMjAxMjAxMTIxNzI3NDRaFw0xMjAxMTIxNzI3NDRaMEgxCzAJBgNV
BAYTAkJNMRkwFwYDVQQKEsBRdW9WYWRRpcyBMAW1pdGVkMRswGQYDVQQDEsJRdW9W
YWRpcyBSb290IENBIDEGRzMwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCa
GMpLlA0ALa8DKYrwD4HIrkWZhR0In6spRlXzL4GtMh6QRr+jhiYaHv5+HBg6XJxg
Fyo6dIMzMH1hVBHL7avg5tKifVrbxi3Cgst/ek+7wrGsxDp3MJGF/hd/aTa/55J
WpzmM+Yklvc/ulsrHHo1wtZn/qtmUlItKGAr79dgw8eTvi02kfn/+NsRE8Scd3bB
rrcCaoF6qUWD4gXmuVbBlDePSHFjIuwXZQeVikvfj8ZaCuWw419eagGrDPmF60Tp
+ARz8un+XJiM9XOva7R+zdRcAitMOeGylZUtQofX1bOQQ7dsE/He3fBE+Ik/0XX1
ksOR1Yqi0JDs3G3eicJlcZaLDQP9nL9bFqyS2+r+eXyt66/3FsvbzSUR5R/7mp/i
Ucw6UwxI5g69ybR2BILmEROfcmMDBOAENisgGQLodKcftslWZvB1JdxnwQ5hYliz
PtGo/KPaHbDRsSNU30R2be1B2MGyIrZTHN81Hdyhdyox5C315eXbyOD/5YDXC2Og
/zOhD7osFRXql7PsrW+8oyWHhqPHWykYTe5hnMz15eWniN9gqRMgeKh0bnpX5UH
oycR7hYQe7xFSkyyBNKr79X9DFHOUGoIMfmR2gyPZFwDwzqLID9ujWc9Otb+fVuI

yV77zGHcizN300QyNqliBIWENieJ0f7OyHj+OsdWwIDAQABo4GwMIGtMA8GA1Ud
EwEB/wQFMAMBAf8wCwYDVR0PBAQDAgEGMB0GA1UdDgQWBbQahGK8SEwzJQTU7tD2
A8QZRtGUazBuBgNVHSMEZzBlgBQahGK8SEwzJQTU7tD2A8QZRtGUa6FJpEcwRTEL
MAkGA1UEBhMCQk0xGTAXBgNVBAoTEFFf1b1ZhZGlzIExpblWl0ZWQxGzAZBgNVBAMT
ElF1b1ZhZGlzIFJvb3QgQ0EgMoICBQkwDQYJKoZIhvcNAQEFBQADggIBAD4KfK2f
BluornFdLwUvZ+YTRYPENvbzwCYMDbVHZF34tHLJRqUDGCdViXh9duqWNIAXINzn
g/iN/Ae42l9NLmeyhP3ZRPx3UIHmfLTJDQtyU/h2BwdBR5YM++CCJpNVjp4iH2Bl
fF/nJrP3MpCYUNQ3cVX2kiF495V5+vgtJodmVjB3pjd4M1IQWK4/YY7yarHvGH5K
WWPKjaJW1acvvFYfzsnB4vsKqBUsfU16Y8Zsl0Q80m/DSHcK+JDSV6IZUaUtl0Ha
B0+pUNqQjZRG4T7wlP0QADj1O+hA4bRuVhogzG9Yje0uRY/W6ZM/57Es3zrWIoZc
hLsib9D45MY56QSPIMO661V6bYcZJPVsAfv4l7CUW+v90m/xd2gNNWQjrLhVoQPR
TUIZ3Ph1WVaj+ahJefivDrkRoHy3au000LYmYjgahwz46P0u05B/B5EqHdZ+XIWD
mbA4CD/pXvk1B+TJYm5Xf6dQlfe6yJvmjqIBxdZmv3lh8zwc4bmCXF2gw+nYSL0Z
ohEUGW6yhhhtoPkg3Goi3XZZenMfvJ2II4pEZXNLxId26F0KCl3GBUzGpn/Z9Yr9y
4aOTHcyKJlOJONDO1w2AFrR4pTqHTI2KpdVGl/IsELm8VCLAABpQ570su9t+Oza
8eOx79+Rj1QqCyXBjhnEUhAFZdWCEOrCMc0u
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUrfC0JFuBiZs18s64KztbpybwdSgwDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFFf1b1ZhZGlzIExpblWl0ZWQxHjAc
BgNVBAMTFVFf1b1ZhZGlzIFJvb3QgQ0EgMiBHMzAeFw0xMjAxMTIxODU5MzJaFw00
MjAxMTIxODU5MzJaMEGxCzAJBgNVBAYTAkNMRkwFwYDVQQKEExBRdW9WYWWRpcyBM
aW1pdGVkMR4wHA YDVQQDExVRdW9WYWWRpcyBSb290IENBIDIgRzRwggliMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQChriWyARjcV4g/Ruv5r+LrI3HimtFhZiFf
qq8nUeVuGxbULX1QsFN3vXg6YOJkApt8hpvWGo6t/x8Vf9WVHhLL5hSEBMHfNrMW
n4rjyduYNM7YMXcoRvynyfDStNVNCXJJ+fKH46nafaF9a7I6JaltUkSs+L5u+9ym
c5GQYyA YDFCDy54ejiK2toIz/pgsIUxNfGhVY7g1gQyJO/Dh4fxaXc6AcW34Sas+
O7q414AB+6XrW7PFXmAqMaCvN+ggOp+oMiwMzAkd056OXbxMmO7FGmh77FOM6RQ1
o9/NgJ8MSPsc9PG/Srj61YxxSscfrf5BmrODXfKEVU+IV0POKa2Mq1W/xPtbAd0j
IaFYAI7D0GoT7RPjEiuA3GfmlbLNHiJuKvhB1PLKFAeNilUSxmn1uIZoL1NesNKq
IcGY5jDjZ1XHm26sGahVpkUG0CM62+tlXSoREfA7T8pt9DTEceT/AFr2XK4jYIVz
8eQqSsWu1ZK7E8EM4DnatDlXtas1qnIhO4M15zHfeiFuuDIIIfR0ykRVKYnLP43eh
vNURG3YBZwjgQQvD6xVu+KQZ2aKrr+InUIYrAoosFCT5v0ICvybIxo/gbjh9Uy3l
7ZizlWNof/k19N+IxWA1ksB8aRxlRbQ694Lrz4EEEVIWFA4r0jyWbYW8jwNkALG
cC4BrTwV1wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQU7edvdlq/YOxJW8ald7tyFnGbxD0wDQYJKoZIhvcNAQELBQAD
ggIBAJHfgD9DCX5xwvfrs4iP4VGyvD11+ShdyLyZm3tdquXK4Qr36LLTn91nMX66
AarHake7kNQIXLJgapDwyM4DYvmL7ftuKtwGTTwpD4kWilhMSA/ohGHqPHKmd+RC
roijQ1h5fq7KpVMNqT1wvSAZYaRsOPxDMuHBR//47PERIjKWnML2W2mWeyAMQ0Ga
W/ZZGYjeVYy3UQt4XAoeo0L9x52ID8DyeAlkVJOviYeIyUqAHerQbj5hLja7NQ4n
lv1mNDthcnPxFlxHBIRJAHpYERAK74X9sbgzdWqTHBLmYF5vHX/JHyPLhGGfHoJE
+V+tYIUkmlKY7VHnoX6XOuYvHxHaU4AshZ6rNRDbll9qxV6XU/IyAgkwo1jwDQHV
csaxfGl7w/U2Rcxhbl5MLMVerugOXou/983g7aEOGzPuVBj+D77vfoRrQ+NwmNtd
dbINWQeFFSM51vHfqSYP1kjHs6Yi9TM3WpVHn3u6GBVv/9YUZINJ0gpnIdsPNWNg
KCLjsZWDzYWm3S8P52dSbrsvhXz1SnPnxT7AvSESBT/8twNJA1vIjebiVDj1eYeM
HVOyToV7BjjHLPj4sHKNJeV3UvQDHEimUF+IIDBu8oJDqz2XhOdT+yHBTw8imoa4

WSr2Rz0ZiC3oheGe7IUlarFsNMkd7EgrO3jtZsSOeWmD3n+M

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGnTCCBIWgAwIBAgICBcYwDQYJKoZIhvcNAQEFBQAwwRTELMAkGA1UEBhMCQ0x
GTAXBgNVBAoTEFF1b1ZhZGlzIExpblWl0ZWQxGzAZBgNVBAMTElF1b1ZhZGlzIFJv
b3QgQ0EgMzAeFw0wNjExMjQxOTExMjNaFw0zMTEwMjQxOTExMjQxOTA2NDRaMEUx
CzAJBgNVBAYTAkNMRkwFwYDVQQKEwBRdW9WYW9WYWRpcyBMAw1pdGVkMRswGQYD
VQQDEwJRdW9WYWRpcyBSb290IENBIDMwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwg
gIKAoICAQDMV0IwVJzmmNPTTe7+7cefQzLKZbPoFog02w1ZkXTPkrgEQK0CSzGrv
I2RaNggDhoB4hp7Thdd4oq3P5kazethq8Jlph+3t723j/z9cI8LoGe+AaJZz3HmDyl
2/7FWeUUrH556VOijKTVopAFPD6QuN+8bv+OPEKhyq1hX51SGyMnzW9os2l2Objy
jPtr7guXd8lyyBTNvijb00BNO/79KDDRMpsMhvVAEVeuxu537RR5kFd5VAYwCdrX
LoT9CabwvVWhDFlaJKjdHkf2mrk7AyxRlIDdLkgbvBNDInIjbC3uBr7E9KsRlOni
27tyAsdLTmZw67mtaa7ONt9XOnMK+pUsvFrGeaDsGb659n/je7Mwpp5ijJUMv7/FF
JuGITfhebtFZFG4ZM2mnO4SJk8RTVROhUXhA+LjJou57ulJCg54U7QVSWllWp5f8n
T8KKdjct5EOE7zelaTfi5m+rJszio+1ga8bxiJTyPbH7pcUsMV8eFLI8M5ud2CEp
ukqdiDtWAEXMJPPGovgc2PZapKUSU60rUqFxmMiMPwJ7Wgic6aIDFUhWMXhOp8q3
crhkODZc6tsgLjoC2SToJyMGf+z0gzskSaHirOi4XCPLArlzW1oUevaPwV/izLmE1
xr/I9A4iLlTlRkT9a6fUg+qGkM17uGcclzuD87nSVL2v9A6wIDAQABo4IBITCCA
ZEwDwYDVR0TAQH/BAUwAwEB/zCB4QYDVR0gBIH2MIH2MIHTBgkrBgEEAb5YAAMwgc
UwgZMGCCsGAQUFBwICMIGGGGoGDQW55IHVzZSBvZiB0aGlzIENlcnRpZmljYXRlIGNvbnN0
aXR1dGVzIGFjY2VwdGFuY2Ugb2YgdGhlIFF1b1ZhZGlzIFJvb3QgQ0EgMyBDZXJ0
aWZpY2F0ZSBQb2xpy3kgLyBDZXJ0aWZpY2F0aW9uIFByYWN0aWNlIFN0YXRlbnVWu
dC4wLQYIKwYBBQUHAgEWTWh0dHA6Ly93d3cucXVvdmFkaXNnbG9iYWwY29tL2Nw
czALBgNVHQ8EBAMCAQYwHQYDVR0OBBYEFPLAE+CCQz777i9nMpY1XNu4ywLQMG4G
A1UdIwRnMGWAFFPLAE+CCQz777i9nMpY1XNu4ywLQoUmkRzBFMQswCQYDVQQGEwJC
TTEZMBcGA1UEChMQUXVvVmFkaXMgTGltZXIzDEBMBkGA1UEAxMSUXVvVmFkaXMg
Um9vdCBDQSAzggIFxjANBgkqhkiG9w0BAQUFAAOCAgEAT62gLEz6wPJv92ZVqyM0
7ucp2sNbtrCD2dDQ4iH782CnO11gUyeim/YlIirnv6By5ZwkajGxkHon24QRiSem
d1o417+shvzuXYO8BsBrd2sPbSQvS3pspweWyuOEn62lix2rFo1bZhfZFvSLgNLd
+LJ2w/w4E6oM3kJpK27zPOuAJ9v1pkQNN1pVWQvVDVJIxa6f8i+AxeoyUDUSly7B
4f/xI4hROJ/yZlZ25w9Rl6VSDE1JUzu2Pb+iSwwQHYaZTKrzchGT5Or2m9qoXadN
t54CrnMAyNojA+j56hl0YgCUyyIgvpsnWbWCar6ZeXqp8kokUvd0/bpO5qgdAm6x
DYBEwa7TlZdfu4V8K5Iu6H6li92Z4b8nby1dqnuH/grdS/yO9SbkbnBCbjPsMZ57
k8HkyWkaPcBrTiJt7qtYtcbQQcEr6k8Sh17rRdhs9ZgC06DYVYoGmRmioHfRMJ6s
zHXug/WwYjnPbFfiTNKRCw51KBuav/0aQ/HKd/s7j2G4aSGWQgRecCocIdiP4b0j
Wy10QJLZYxkNc91pvGJHvOB0K7Lrfb5BG7XARsWhlStfTsEokt4YutUqKLsRixeT
mJlglFwjz1onl14LBQaTNx47aTbrqZ5hHY8y2o4M1nQ+ewkk2gF3R8Q7zTSMmfXK
4SVhM7JZG+Ju1zdXtg2pEto=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUlvWbAiin23r/1aOp7r0DoM8Sah0wDQYJKoZIhvcNAQEL
BQAwwSDELMAkGA1UEBhMCQ0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpblWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMyBHMzAeFw0wMjAxMTIyMDI2MzJaFw00

MjAxMTIyMDI2MzJaMEgx CzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
aW1pdGVkMR4wHA YDVQQDExVRdW9WYWRpcyBSb290IENBIDMgRzMwggLiMA0GCSqG
SIB3DQEB AQUAA4ICDwAwggIKAoICAQCzyw4QZ47qFJenMioKVjZ/aEzHs286IxSR
/xl/pcqs7rN2nXrpixurazHb+gtTTK/FpRp5PIpM/6zJd5O2YIyC0TeytuMrKNu
FoM7pmRLMon7FhY4futD4tN0SsJiCnMK3UmzV9KwCoWdcTzeo8vAMvMBOSBDGzXR
U7Ox7sWtAYI+FrUoRqHe6okJ7UO4BUAKhvVZR74bbwEhELn9qdIoyhA5CcoTNs+c
ra1AdHkrAj80//ogaX3T7mH1urPnMNA3I4ZyYUUpSFlob3emLoG+B01vr87ERROR
FHAGjx+f+IdpsQ7vw4kZ6+ocYfx6bIrc1gMLnia6Et3UVDmrJqMz6nWB2i3ND0/k
A9HvFZcba5DFApCTZgIhsUfei5pKgLIvJ7WiL8DWM2fafsSntARE60f75li59wzw
eyuxwHApw0BiLTtIadwjPEjrewl5qW3aqDCYz4ByA4imW0aucnl8CAMhZa634Ryl
sSqIMd5mBPfAdOhx3v89WcyWJhKLhZVXGqtrdQtEPREoPHtht+KPZ0/17DxMYIBp
VzgeAVuNVEjH38DMdyM0SXV89pgR6y3e7UEuFAUCf+D+IOs15xGsIs5XPd7JMG0Q
A4XN8f+MFRXBsj6IbGB/kE+V9/YtrQE5BwT6dYB9v0lQ7e/JxHwc64B+27bQ3RP+
ydOc17KXqQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQUxhfQvKjqAkPyGwaZXSuQILnXnOQwDQYJKoZIhvcNAQELBQAD
ggIBADRh2Va1EodVTd2jNTFGu6QHcrxfYWLopfsLN7E8trP6KZ1/AvWkyaiTt3px
KGmPc+FSkNrVvjrlt3ZqVoAh313m6Tqe5T72omnHKgqwGEfcIHB9UqM+WXzBusnI
FUBhynLWcKzSt/Ac5IYP8M7vaGPQtSCKFWGafoaYtMnCdvvMujAWzKNhxnQT5Wvv
oxXqA/4Ti2Tk08HS6IT7SdEQTXlm66r99I0xHnAUrdzeZxNMGRVhvLfZkXdxGYFg
u/BYpbWcC/ePIIUnwEsBbTuZDdQdm2NnL9DuDcpmvJRPpq3t/O5jrFc/ZSXPsoaP
0Aj/uHYUbt7IJ+yreLVtubY/6CD50qi+YUbkH4yE8/nxoGibIh6BJpsQBjFfxwAYf
3KDTuVan45gtf4Od34wrnDKOMpTwATwiKp9Dwi7DmDkHOHv8XgBCH/MyJnmDhPbl
8MFREsALHgQjDFSITC9JxUrRtm5gDWv8a4uFJGS3iQ6rJUdbPM9+Sb3H6QrG2vd+
DhcI00iX0HGS8A85PjRqHH3Y8iKuu2n0M7SmsFXRDw4m6Oy2Cy2nhTXN/VnIn9HN
PlopNLk9hM6xZdRZkZFWdSHBd575euFgndOtBBj0fOtek49TSilp+EgrPk2GrFt/
ywaZWWDYWGwVjUTR939+J399roD1B0y2PpxxVJkES/1Y+Zj0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICIDCCAhqgAwIBAgIILCmcWxbtBZUwCgYIKoZIzj0EAwIwfwELMAkGA1UEBhMC
VVMxMjE2MzJaMEgx CzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
U0wgQ29ycG9yYXRpb24xNDAYBgNVBAMMK1NTTC5jb20gRVYgUm9vdCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eSBFQ0MwHhcNMTYwMjE2MzJaMEgx CzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
NTIzWjB/MQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVG94YXN0aWZpY2F0aW9uIEF1dGhvcml0eSBFQ0MwHhcNMTYwMjE2MzJaMEgx CzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
dXN0b24xGDAWBgNVBAoMD1NTTCBDb3Jwb3JhdGlvdjE0MDIGA1UEAwU1NMLmNv
bSBFViBSb290IENlcuRzZmljYXRpb24xNDAYBgNVBAMMK1NTTC5jb20gRVYgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eSBFQ0MwHhcNMTYwMjE2MzJaMEgx CzAJBgNVBAYTAkJNMRkwFwYDVQQKExBRdW9WYWRpcyBM
AgEGBSuBBAIAiA2IABKoSR5CYG/vvw0AHgyBO8TCCogbR8pKGYfL2IWjKAMTH6kMA
VIbc/R/fALhBYlzcBYY3h+Z1MzFB8giH2EWB1E9fVwHU+M1OIzFzZ/ZLg1Kthku
WnBaBu2+8KGwytAJKaNjMGEwHQYDVR0OBByEFFvKXuXe0oGqzagtZFG22XKbl+ZP
MA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUW8pe5d7SgarNqC1kUbbZcpuX
5k8wDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMCA2gAMGUCMQCK5kCJN+vp1RPZ
ytRrJPOwPYdGWBrsd9v+1a6cGvHOMzosYxPD/fxZ3YOg9AeUY8CMD32IygmTMZg
h5Mmm7I1HrW9zzRHM76JTyMGoEVW/MSD2zuZYrJh6j5B+BimoxcSg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF6zCCA9OgAwIBAgIIvYrYpzTS8ePYwDQYJKoZIhvcNAQELBQAwwYIxCzAJBgNV
BAYTAIVTMQ4wDAYDVQQIDAVUZXhhczEQMA4GA1UEBwwHSG91c3RvbjEYMBYGA1UE
CgwPU1NMIENvcnBvcnF0aW9uMTcwNQYDVQQDDC5TU0wuY29tIEVWIFJvb3QgQ2Vy
dGlmaWNhdGlubiBBdXR0b3JpdHkgUINBIFlyMB4XDTE3MDUzMTE4MTQzN1oXDTQy
MDUzMTE4MTQzN1owgYIxCzAJBgNVBAYTAIVTMQ4wDAYDVQQIDAVUZXhhczEQMA4G
A1UEBwwHSG91c3RvbjEYMBYGA1UECgwPU1NMIENvcnBvcnF0aW9uMTcwNQYDVQQD
DC5TU0wuY29tIEVWIFJvb3QgQ2VyY29tIEVWIFJvb3QgQ2VyY29tIEVWIFJvb3QgQ2Vy
MIICijANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEajZlQOHWTcDXtOlG2mvq
M0fNTPi9fb69LT3w23jhhqXZuglXaO1XPqDQCEGD5yhBJB/jchXQARr7XnAjssuf
OePPxU7Gkm0mxnu7s9onnQqG6YE3Bf7wcXHswxzpY6IXFJ3vG2fThVUCAtZJycxa
4bH3bzKfydQ7iEGonL3Lq9ttewkfokxykNorCPzPPFTOZw+oz12WGQvE43LrrdF9
HSfvkusQv1vrO6/PgN3B0pYEW3p+pKk8OHakYo6gOV7qd89dAFmPziw+B6KjBSYR
aZfqhbcPlgtLyEDhULouisv3D5oi53+aNxPN8k0TayHRwMwi8qFG9kRpnMphNQC
aB9ZhCBHqurj26bNg5U257J8UZslXWNvNh2n4ioYSA0e/ZhN2rHd9NCSFg83XqpyQ
Gp8hLH94t2S42Oim9HizVcuE0jLEeK6jj2HdzghTreyI/BXkmg3mnp3zkyPuBQV
PWKchjgGAGYS5F12WIPAApiiECtoRHuOec4zSnaqW4EWG7WK2NAAe15itAnWhmMO
pgWVSbooi4iTsjQc2KRvbrcc0N6ZVTsj9CLg+SImJuwgUHfbSguPvuUCYHBBXtSu
UDkiFCbLsjtZdFVHB3mBOagwE0TIBIqulhMIQg+5U8Sb/M3kHN48+qvWBkofZ6aY
MBzdLNvcGJVXZsb/XItW9XcCAwEAAANjMGEwDwYDVR0TAQH/BAUwAwEB/zAfBgNV
HSMGEGAwwBT5YLvU49U09rj1BoAlp3PbRmmonjAdBgNVHQ4EFgQU+WC71OPVNP
a49QaAJadz20ZpqJ4wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEBCwUAA4ICAQBW
s47LCp1Jjr+kxJG7ZhcFUZh1++VQLHqe8RT6q9OKPv+RKY9ji9i0qVQBDb6Thi/5
Sm3HXvVX+cpVHBK+Rw82xd9qt9t1wkclF7nxY/hoLVUE0fKNsKTPvDxeH3jnpaAg
cLAExbf3cqfeIg29MyVGjGSSJuM+LmOW2puMPfGYCdcDzH2GguDKBAdRUNf/ktUM
79qGn5nX67evaOI5JpS6aLe/g9Pqemc9YmeuJeVy6OLk7K4S9ksrPJ/psEDzOFsZ
/bdoyNrGj1E8svuR3Bznm53htw1yj+KkxKl4+esUrMZDBcJlOSgYAsOCsp0FvmXt
ll9ldDz7CTUue5wT/RsPXcdtGtpWD8w74a8CLyKsRspGPKAcTNZEtF4uXBVmcEm
Kf7GUmG6sXP/wwyc5WxqlD8UykAWIYTzWamsX0xhk23RO8yilQwipmdnRC652dKK
QbNmC1r7fSOl8hqw/96bg5Qu0T/fkreRrwU7ZcegbLHNYhLDkBVjJc40vG93drEQ
w/cFGsDWR3RiSBd3kmmQYRzelYB0VI8YHMPzA9C/pEN1hlMYegouCRw2n5H9gooi
S9EOUCXdywMMF8mDAAhONU2Ki+3wApRmLER/y5UnlhetCTCstnEXbosX9hwJ1C07
mKVx01QT2WDz9UtmT/rx7iASjbSsV7FFY6GsdqnC+w==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICjTCCAHSgAwIBAgIIdebFy8FoW6gCgYIKoZIzj0EAwIwFDELMakGA1UEBhMC
VVMxMjE4MTQzN1owgYIxCzAJBgNVBAYTAIVTMQ4wDAYDVQQIDAVUZXhhczEQMA4GA1UEBwwHSG91c3RvbjEYMBYGA1UECgwPU1NMIENvcnBvcnF0aW9uMTcwNQYDVQQDDC5TU0wuY29tIEVWIFJvb3QgQ2Vy
dGlmaWNhdGlubiBBdXR0b3JpdHkgUINBIFlyMB4XDTE3MDUzMTE4MTQzN1oXDTQy
MDUzMTE4MTQzN1owgYIxCzAJBgNVBAYTAIVTMQ4wDAYDVQQIDAVUZXhhczEQMA4G
A1UEBwwHSG91c3RvbjEYMBYGA1UECgwPU1NMIENvcnBvcnF0aW9uMTcwNQYDVQQD
DC5TU0wuY29tIEVWIFJvb3QgQ2VyY29tIEVWIFJvb3QgQ2VyY29tIEVWIFJvb3QgQ2Vy
MIICijANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEajZlQOHWTcDXtOlG2mvq
M0fNTPi9fb69LT3w23jhhqXZuglXaO1XPqDQCEGD5yhBJB/jchXQARr7XnAjssuf
OePPxU7Gkm0mxnu7s9onnQqG6YE3Bf7wcXHswxzpY6IXFJ3vG2fThVUCAtZJycxa
4bH3bzKfydQ7iEGonL3Lq9ttewkfokxykNorCPzPPFTOZw+oz12WGQvE43LrrdF9
HSfvkusQv1vrO6/PgN3B0pYEW3p+pKk8OHakYo6gOV7qd89dAFmPziw+B6KjBSYR
aZfqhbcPlgtLyEDhULouisv3D5oi53+aNxPN8k0TayHRwMwi8qFG9kRpnMphNQC
aB9ZhCBHqurj26bNg5U257J8UZslXWNvNh2n4ioYSA0e/ZhN2rHd9NCSFg83XqpyQ
Gp8hLH94t2S42Oim9HizVcuE0jLEeK6jj2HdzghTreyI/BXkmg3mnp3zkyPuBQV
PWKchjgGAGYS5F12WIPAApiiECtoRHuOec4zSnaqW4EWG7WK2NAAe15itAnWhmMO
pgWVSbooi4iTsjQc2KRvbrcc0N6ZVTsj9CLg+SImJuwgUHfbSguPvuUCYHBBXtSu
UDkiFCbLsjtZdFVHB3mBOagwE0TIBIqulhMIQg+5U8Sb/M3kHN48+qvWBkofZ6aY
MBzdLNvcGJVXZsb/XItW9XcCAwEAAANjMGEwDwYDVR0TAQH/BAUwAwEB/zAfBgNV
HSMGEGAwwBT5YLvU49U09rj1BoAlp3PbRmmonjAdBgNVHQ4EFgQU+WC71OPVNP
a49QaAJadz20ZpqJ4wDgYDVR0PAQH/BAQDAgGGMA0GCSqGSIb3DQEBCwUAA4ICAQBW
s47LCp1Jjr+kxJG7ZhcFUZh1++VQLHqe8RT6q9OKPv+RKY9ji9i0qVQBDb6Thi/5
Sm3HXvVX+cpVHBK+Rw82xd9qt9t1wkclF7nxY/hoLVUE0fKNsKTPvDxeH3jnpaAg
cLAExbf3cqfeIg29MyVGjGSSJuM+LmOW2puMPfGYCdcDzH2GguDKBAdRUNf/ktUM
79qGn5nX67evaOI5JpS6aLe/g9Pqemc9YmeuJeVy6OLk7K4S9ksrPJ/psEDzOFsZ
/bdoyNrGj1E8svuR3Bznm53htw1yj+KkxKl4+esUrMZDBcJlOSgYAsOCsp0FvmXt
ll9ldDz7CTUue5wT/RsPXcdtGtpWD8w74a8CLyKsRspGPKAcTNZEtF4uXBVmcEm
Kf7GUmG6sXP/wwyc5WxqlD8UykAWIYTzWamsX0xhk23RO8yilQwipmdnRC652dKK
QbNmC1r7fSOl8hqw/96bg5Qu0T/fkreRrwU7ZcegbLHNYhLDkBVjJc40vG93drEQ
w/cFGsDWR3RiSBd3kmmQYRzelYB0VI8YHMPzA9C/pEN1hlMYegouCRw2n5H9gooi
S9EOUCXdywMMF8mDAAhONU2Ki+3wApRmLER/y5UnlhetCTCstnEXbosX9hwJ1C07
mKVx01QT2WDz9UtmT/rx7iASjbSsV7FFY6GsdqnC+w==

cmFqb3dhIEl6YmEgUm96bGljemVuaW93YSBTLkEuMRgwFgYDVQQDDA9TWkFGSVIg
Uk9PVCBDQTIwggEiMA0GCSqSIlb3DQEBAAUAA4IBDwAwggEKAoIBAQC3vD5QqEvN
QLXOYeeWyrSh2gwisPq1e3YAd4wLz32ohswmUeQgPYUM1lj5/QqGJ3a0a4m7utT
3PSQ1hNKDJA8w/Ta0o4NkjrcsbH/ON7Dui1fgLkCvUqdGw+0w8LBZwPd3BucPbOw
3gAeqDRHu5rr/gSUVtaE2g0gv/pby6kWIK05YO4vdbbnl5z5Pv1+TW9NL++IDWr6
3fE9biCloBK0TXC5ztdyO4mTp4CEHCdJckm1/zuVnsHMyAHs6A6KCpbns6aH5db5
BSsNl0BwPLqsdVqc1U2dAgrSS5tmS0YHF2Wtn2yIANwiieDhZNRnvDF5YTy7ykHN
XGoAyDw4jlivAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMB0GA1UdDgQWBBQuFqlKGLXLzPVvUPMjX/hd56zwyDANBgkqhkiG9w0BAQSF
AAOCAQEAtXP4A9xZWx126aMqe5Aosk3AM0+qmrHUuOQn/6mWmc5G4G18TKI4pAZw
8PRBEew/R40/cof5O/2kbytTAOD/OblqBw7rHRz2onKQy4I9EYKL0rufKq8h5mOG
nXkZ7/e7DDWQw4rtTw/1zBLZpD67oPwglV9PJi8RI4NOdQcPv5vRtB3pEAT+ymCP
oky4rc/hkA/NgrHXXu3UNLUYfrVFdvXn4dRVoul4+vJhaAlIDf7js4MNIThPIGy
d05DpYhfhmehPea0XGG2Ptv+tyjFogeutcrKjSoS75ftwjCkySp6+/NNI XuZmZSg
LvWpCz/UXeHPhJ/iGcJfitYgHuNztw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICOjCCAcGgAwIBAgIQQvLM2htpN0RfFf51KBC49DAKBggqhkiOPQQDAzBfMQsw
CQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1T
ZWN0aWdvIFB1YmXpYyBTZXJ2ZXIgaXQXV0aGVudGljYXRpb24gUm9vdCBFN DYwHhcN
MjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEYMBYG
A1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1YmXpYyBT
ZXJ2ZXIgaXQXV0aGVudGljYXRpb24gUm9vdCBFN DYwdjAQBgcqhkiOPQIBBgUrgQQA
IgNiAAR2+pmpbiDt+dd34wc7qNs9Xzjoq1WmVk/WSOrsfy2qw7LFeeyZYX8QeccC
WvkEN/U0NSt3zn8gj1KjAIns1aeibVvjS5KToID1AZTc8GgHHs3u/iVStSBDHBv+
6xnOQ6OjQjBAMB0GA1UdDgQWBBTRItPmWfFLXyY4qp3W7usNw/upYTAOBgNVHQ8B
Af8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkiOPQQDAwNnADBKAjAn7qRa
qCG76UeXlImldCBteU/IvZNeWBj7LRoAasm4PdCkT0RHlAFWovgzJQxC36oCMB3q
4S6ILuH5px0CMk7yn2xVdOOurvulGu7t0vzCAxHrRVxgED1cf5kDW21USAGKcw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFijCCA3KgAwIBAgIQdY39i658BwD6qSWn4cetFDANBgkqhkiG9w0BAQwFADBf
MQswCQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQD
Ey1TZWN0aWdvIFB1YmXpYyBTZXJ2ZXIgaXQXV0aGVudGljYXRpb24gUm9vdCBSNDYw
HhcNMjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEY
MBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1YmXp
YyBTZXJ2ZXIgaXQXV0aGVudGljYXRpb24gUm9vdCBSNDYwgglMA0GCSqGSIlb3DQEB
AQUAA4ICDwAwggIKAoICAQCTvtU2UnXYASOGHEdCSe5jtrch/cSV1UgrJnwUUXDa
ef0rty2k1Cz66jLdScK5vQ9IPXtamFSvn10xdE8H/Fah3aTPaE8bEmNtJZIMKpnz
SDBh+oF8HqclStw+KxwfGExxqjWMrfhu6DtK2eWUAtaJhBOqbchPM8xQljeSM9xf
iOefVNI8JhD1mb9nxc4Q8UBUQvX4yMPFF1bFOdLvt30yNoDN9HWOaEhUTCDSG3X
ME6WW5HwcCSrv0WBZEMNvSE6Lzzpng3LILVCJ8zab5vuZDCQOc2TZYehMbUjUDM3
luM47fgxMMxF/mL50V0yeUKH32rMVhlAtC6qu/m1dkmU8Sf4kaWD5QazYw6A3OAS
VYCmO2a0OYctyPDQ0RTp5A1NDvZdV3LFOxxHVP3i1fuBYYzMTYCQNFu31xR13NgE

SJ/AwSiItOkcyqex8Va3e0IMWeUgFaiEAin6OJRpmkkGj80feRQXEgyDet4fsZfu
+Zd4KKTIRJLpfSYFplhym3kT2BFfrsU4YjRosoYwjviQYZ4ybPUHNS2iTg7sijbt
8uaZFURww3y8nDnAtOfR94MI1fZEoDlSfB1D++N6xybVCi0ITz8fAr/73trdf+L
HaAZBav6+CuBQug4urv7qv094PPK306Xlynt8xhW6aWwRL3DkJiy4Pmi1KZHQ3xt
zwIDAQABo0IwQDAdBgNVHQ4EFgQUVnNYZJX5khqwEioEYnmhQBWIIUkwDgYDVR0P
AQH/BAQDAgGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBAQDggIBAC9c
mTz8Bl6MlC5w6tIyMY208FHVvArzZJ8HXtXBc2hkeqK5Duj5XYUtqDdFqij0lgVQ
YKIJfp/imTYpEORHap1VIDzYm/EDMrraQKFz6oOht0SmDpkBm+S8f74TlH7Kph52
gDY9hAaLMyZlbcP+nv4fjFg4exqDsQ+8FxG75gbMY/qB8oFM2gsQa6H61SilzwZA
Fv97fRheORKkU55+MkIQpiGRqRxoF3yEvJ+M0ejf5lG5Nkc/kLnHvALcWxxPDkjB
JYOcCj+esQMzEhonrPcibCTRAUH4WAP+JWgiH5paPHxsnnVI84HxZmdutILA7rpX
DhjvLpr3Etiga+kFpaHpaPi8TD8SHkXoUsCjvxInebnMMTzD9joiFgOgyY9mpFui
TdaBJQbpdqQACj7LzTWb4OE4y2BThihCQRxEV+ioratF4yUQvNs+ZUH7G6aXD+u5
dHn5HrwdVw1Hr8Mvn4dGp+smWg9WY7ViYG4A++MnESLn/pmPNPW56MORcr3Ywx65
LvKRRFHQV80MNNVIIb/bE/FmJUNS0nAiNs2fxBx1IK1jcmMGDw4nztJqDby1ORrp
0XZ60Vzk50lJLVU3aPaAOpG+VBeHVOmmJ1CJeyAvP/+oYtKR5j/K3tJPsmPmAY
QqsZKbrAKbkTidOIjIjBO8n9pu0f9GBj39ItVQGL

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDCjCCAlqgAwIBAgIUZvnHwa/swlG07VOX5uaCwysckBYwDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMdGQuMR0wGwYDVQQDEExRTZWV1cmVtaWduIFJvb3QgQ0ExMjAeFw0yMDA0MDgw
NTM2NDZaFw00MDA0MDgwNTM2NDZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD
eWJlcnRydXN0IEphcGFuIEVnLiwgTHRkLjEEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC6OcE3emhF
KxS06+QT61d1I02PJC0W6K6OyX2kVzsQdiUz2zqMoqUm048luT9Ub+ZyZN+v/mt
p7JIKwccJ/VMvHASd6SFVLX9kHrko+RRWAPNEHl57muTH2SOa2SroxPjcf59q5zd
J1M3s6oYwlkm7Fsf0uZlfo+TvdhYXAvaA42VvPMfKWeP+bl+sg779XSVOKik71gur
FzJ4pOE+IEa+Ym6b3kaosRbnhW70CEBFEEaCeVESE99g2zvVQR9wsMJvuwPWW0v4J
hscGwa5Pro4RmHvzC1KqYiaqId+OJTN5lxZJfU+1UefNzFJM3IFTQy2VYzxV4+K
h9GtxRESOaCtAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQD
AgEGMB0GA1UdDgQWBBRXNPN0zwRL1SXm8UC2LEzZLemgrTANBgkqhkiG9w0BAQsF
AAOCAQEAPrvbFxbS8hQBICw4g0utvsqFepq2m2um4fylOqyttCg6r9cBg0krY6Ld
mmQOmFvx3Y67ilQlUoT865AQ9tPkbeGGuWAtEGBpE/6aouIs3YIcipJQMPTw4WJ
mBClnW8Zt7vPemVV2zfrPIpyMpcemik+rY3moxtt9XUa5rBouVui7mlHJzWhhpmA
8zNL4WukJsPvdFlseqJkth5Ew1DgDzk9qTPxpFpSvWKErI4cq1avTc7bgoitPQV
55FYxTpE05Uo2cBl6XLK0A+9H7MV2anjPecJnuDLN/v9vZfVvhgaaaI5gdka9at/
yOPiZwud9AzqVN/Ssq+xlVeg37xEHA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFCjCCA1qgAwIBAgIUZNtaDCBO6Ncpd8hQJ6JaJ90t8sswDQYJKoZIhvcNAQEM
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMdGQuMR0wGwYDVQQDEExRTZWV1cmVtaWduIFJvb3QgQ0ExNDAAeFw0yMDA0MDgw
NzA2MTlaFw00NTA0MDgwNzA2MTlaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD

eWJlcnRydXN0IEphcGFuIENvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTQwgglIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDF0nqh1oq/
FjHqMNE6lPxauG4iwWL3pwon71D2LrGeaBLwbCRjOfHw3xDG3rdSINVSW0KZnvOg
vllfX8xbacuUKLBl422+JX1sLrcneC+y9/3OPJH9aaakpUqYllQC6KxNedlsmGy
6pJxaeQp8E+BgQQ8sqVb1MWoWWd7VRxJq3qdwdzTe/NCcLEVxLbAQ4jeQkHO6Lo
/IrPj8BGJJw4J+CDnRugv3gVEOuGTgpa/d/aLIJ+7sr2KeH6caH3iGicnPCNvg9J
kdjqOvn90Ghx2+m1K06Ckm9mH+Dw3EzsythqunQG+bOEKJTRX45zGRBDauVwpcAQ
0BB8b8VYSbSwbprafZX1zNoCr7gsfXmPvkPx+SgojQID+Ajda8iLLCSxjVIHvXib
y8posqTdDEx5YMaZ0ZPxMB0H064iwurO8YQJzOAUbn8/ftKChazcqRZOhaBgy/ac
18izju3Gm5h1DVXoX+WVWiwKkrkMpKBGk5hIwAUt1ax5mnXkvpXYvHUC0bcl9eQjs
0Wq2XSqypWa9a4X0dFbD9ed1Uigsp9mR6XU/v6eVL9lfgHwMI+INpyiUBzuOIAB
SMbHdPTGrMNASRZhdCyvjG817XsYAFs2PJxQDcqSMxDxJklt33UkN4li1+iW/RVL
ApY+B3KVfqs9TC7XyvDf4Fg/LS8EmjijAQIDAQAB0IwQDAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUBP0jCl4oaTeqYR3r6/wtbyPk
86AwDQYJKoZIhvcNAQEMBQADggIBAJaAcgkGfpzMkwQWu6A6jZJOtxEaCnFxE0E
rX+IRVAQZk5KQaID2RFPeje5S+LGjzJmdSX7684/AykmjbgWHfYfM25I5uj4V7Ib
ed87hwriZLoAymzvftAj63iP/2SbNDefNWwipAA9EiOWWF3KY4fGoweITedpdopT
zfFP7ELyk+OZpDc8h7hi2/DsHzc/N19DzFGdtfCXwreFamgLRB7IUe6TzktuhsHS
DCRZNhqfLJGP4xjblJUK7ZGqDpncllPjYYPGFrojutzdFhrGe0K22VoF3Jpf1d+4
2kd92jjbrDnVHmtsKheMYc2xbXIBw8MgAGJoFjHVdqqGuw6qnsb58Nn4DSEC5MUo
FlkRudlpcyqSeLiSV5sI8jrlL5WwWLdrIBRtFO8KvH7YVdiI2i/6GaX7i+B/OfVy
K4XELKzvGUWSTLNB9xNH27SgrNcmvMSZ4PPmz+Ln52kuaiWA3rF7iDeM9ovnhp6
dB7h7sxaOgTdsxoEqBRjrLdHEoOabPXm6RUVkRqEGQ6UROcSjiVbgGcZ3GOTEAtl
Lor6CZpO2oYofaphNdgOpygau1LgePhsumywbmHXumZNTfxPWQrqaA0k89jL9WB
365jJ6UeTo3cKXhZ+PmhIIynJkBugnLNeLLIjzwec+fBH7/PzqUqm9tEZDKgu39c
JRNIx+S

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICIZCCAamgAwIBAgIUfHxHw9hJp75pDIqI7fBw+d23PocwCgYIKoZIzj0EAwMw
UTELMAKGA1UEBhMCSlAxIzAhBgNVBAoTGKn5YmVydHJ1c3QgSmFwYW4gQ28uLCBM
dGQuMR0wGwYDVQQDEExRTZWV1cmVtaWduIFJvb3QgQ0ExNTAeFw0yMDA0MDgwODMy
NTZaFw00NTA0MDgwODMyNTZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKEExpDeWJl
cnRydXN0IEphcGFuIENvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBSb290
IENBMTUwdjAQBgqhkJOPQIBBgUrgQQAIGNiAAQLUHSNZDKZmbPSYAi4Io5GdCx4
wCtELW1fHcmuS1Iggz24FG1Th2CeX2yF2wYUleDHKP+dX+Sq8bOLbe1PL0vJSpSR
ZHX+AezB2Ot6IHhWGENfa4HL9rzatAy2KZMIaY+jQJBAMA8GA1UdEwEB/wQFMAMB
Af8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBbTrQciu/NWeUUj1vYv0hyCTQSVt
9DAKBggqhkJOPQDAwNoADBlAjEA2S6Jfl5OpBEHvVnCB96rMjhTKkZEBhd6zlHp
4P9mLQlO4E/0BdGF9jVg3PVys0Z9AjBEmEYagoUeYWmJSwdLZrWeqrqgHkHZAXQ6
bkU6iYAZezKYVWOr62Nuk22rGwlgMU4=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDuDCCAqCgAwIBAgIQDPCOXAgWpa1Cf/DrJxhZ0DANBgkqhkiG9w0BAQUFADBI
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x

FzAVBgNVBAMTDiNIY3VyZVRydXN0IENBMB4XDTA2MTEwNzE5MzExOFoXDTI5MTIz
MTE5NDA1NVowSDELMaKGA1UEBhMCVVMxIDAeBgNVBAoTF1NIY3VyZVRydXN0IENv
cnBvcnF0aW9uMRcwFQYDVQQDEw5TZWN1cmVUcnVzdCBDQTCCASlwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAKukgeWVzfX2FI7CT8rU4niVWJxB4Q2ZQCQXOEz
Zum+4YOvYlyJ0fwkW2Gz4BERQRwdbvC4u/jep4G6pkjGnx29vo6pQT64lOOpGtSO
0gMdA+9tDWccV9cGrcrI9f4Or2YlSASWC12juhbDCE/RRvgUXPLIXgGZbf2IzIao
wW8xQmxSPmjL8xk037uHGFAJsTQ3MBv396gwpEWoGQRS0S8Hvbn+mPeZqx2pHGj
7DaUaHp3pLHnDi+BeuK1cobvomuL8A/b01k/unK8RCSc43Oz969XL0Imnal0ugBS
8kvNU3xHCzaFDmapCJcWNffBZveA4+1wVMeT4C4oFVmHursCAwEAAaOBnTCBmjAT
BgkrBgEEAYI3FAIEBh4EAEMAQTALBgNVHQ8EBAMCAYYwDwYDVR0TAAQH/BAUwAwEB
/zAdBgNVHQ4EFgQUQjK2FvoE/f5dS3rD/fdMQB1aQ68wNAYDVR0fBC0wKzApoCeg
JYYjaHR0cDovL2Nybc5zZWN1cmV0cnVzdC5jb20vU1RDQS5jcmwwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADggEBADdtT0rhWDpSclu1pqNlGKa7UTt3
6Z3q059c4EVlew3KW+JwULKUBRSuSceNQQcSc5R+DCMh/bwQf2AQWnL1mA6s7Ll/
3XpvXdMc9P+IBWlCqQVxyLesJugutIxq/3HcuLHfmbx8IVQr5Fiiu1cprp6poxkm
D5kuCLDv/WnPmRoJjeOnnyvJNjR7JLN4TJUXpAYmHrZkUjZfYGfZnMUFdAvnZyPS
CPyI6a6Lf+Ew9Dd+/cYy2i2eRDAwbO4H3tI0/NL/QPZL9GZGBISm8jIKYyYwa5vR
3ItHuuG51WLQoqD0ZwV4KWMabwTW+MZMo5qxN7SN5ShLHZ4swrhovO0C7jE=
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDvDCCAqSgAwIBAgIQB1YipOjUiolN9BPI8PjqpTANBgkqhkiG9w0BAQUFADBK
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x
GTAXBgNVBAMTEFNiY3VyZSBHbG9iYWwgQ0EwHhcNMdYxMTA3MTk0MjI4WWhcNMjkx
MjMxMTk1MjA2WjBKMqSwCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3Qg
Q29ycG9yYXRpb24xGTAXBgNVBAMTEFNiY3VyZSBHbG9iYWwgQ0EwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvNS7YrGxVaQZx5RNoJLNP2MwhR/jxYdIJ
iQPpvepeRlMJ3Fz1Wuj3RSoC6zFh1yKzTM7HfAo3fg+6MphHZevj8fcyTiW89sa
/FHtaMbQbqR8JNGuQsiWUGMu4P51/pinX0kuleM5M2SOHqRfkNjNPLLZ/kG5VacJ
jnIFHovdRIWCQtBJwB1g8NEXLJXr9qXBkqPFwqcIYA1gBBCWeZ4WNOaptvolRTnI
HmX5k/Wq8VLcmZg9pYYaDDUz+kulBAYVHDGA76oYa8J719rO+TMg1fW9ajMtgQT7
sFzUnKPiXB3jqUJ1XnvUd+85VLrJChgbEplJL4hL/VBi0XPnj3pDAgMBAAGjgZ0w
gZowEwYJKwYBBAGCNxQCBAYeBABDAEEwCwYDVR0PBAQDAgGMAA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEFK9EBMJbfiD2045AuzshHrmzsmkMDQGA1UdHwQtMCsw
KaAnoCWGI2h0dHA6Ly9jcmwuc2VjdXJldHJ1c3QuY29tL1NHQ0EuY3JsMBAGCSsG
AQQBgjcVAQQDAgEAMA0GCSqGSiB3DQEBBQUAA4IBAQBjGghAfaReUw132HquHw0L
URYD7xh8yOOvaliTFGCRsoTciE6+OYo68+aCiV0BN7OrJKQVDpI1WkpEXk5X+nXO
H0jOZvQ8QCaSmGwb7iRGDBezUqXbpZGRzzfTb+cnCDpOGR86p1hcF895P4vkp9Mm
I50mD1hp/Ed+stCNI5O/KU9DaXR2Z0vPB4zmAve14bRDtUstFJ/53CYNv6ZHdAbY
iNE6KTCEztI5gGlbqMdXSbxqVVFuFuq+NQfk1XWYN3kwFNspnWzFacxHVaIw98xc
f8LDmBxrThaA63p4ZUWiABqvDA1VZDRIuJK58bRQKfJPIx/abKwfROHdI3hRW8cW
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICODCCAb6gAwIBAgIJANZdm7N4gS7rMAoGCCqGSM49BAMDMGExCzAJBgNVBAYT
AkpQMSUwIwYDVQQKExxTRUNPTSBUcnVzdCBTeXN0ZW1zIENPLixMVEQuMSswKQYD

VQQDEyJTZWn1cml0eSBDb21tdW5pY2F0aW9uIEVDQyBSb290Q0ExMB4XDTE2MDYx
NjA1MTUyOFoXDTM4MDExODA1MTUyOFowYTELMaKGA1UEBhMCSlAxAjBjBgNVBAoT
HFNFQ09NIFRydXN0IFN5c3RlbXMgQ08uLExURC4xKzApBgNVBAMTIINiY3VyaXR5
IENvbW11bmljYXRpb24gRUNDIjFvbn3RDQTEwdjAQBgcqhkJOPQIBBgUrgQQAIGNi
AASkpW9gAwPDvTH00xecK4R1rOX9PVdu12O/5gSJko6BnOPpR27KkBLIE+Cnnfdl
dB9sELL05OnvbYUymUSxXv3MdhDYW72ixvnWQuRXdtyQwjWpS4g8EkdtXP9JTxpK
ULGjQjBAMB0GA1UdDgQWBBSGHOf+LaVKiwj+KBH6vqNm+GBZLzAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkJOPQQDAwNoADBlAjAVXUI9/Lbu
9zuxNuie9sRGKEkz0FhDKmMpzE2xtHqiuQ04pV1IKv3LsnNdo4gIxwwCMQDAqy0O
be0YottT6SXbVQjgUMzfRGEWgqtJsLKB7HOHeLRMsmIbEvoWTSVLY70eN9k=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDdzCCAl+gAwIBAgIBADANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEWJKUDEl
MCMGA1UEChMcU0VDT00gVHJ1c3QgU3lzdGVtcyBDTy4sTFRELjEnMCUGA1UECXM
U2VjdXJpdHkgQ29tbXVuaWNhdGlvbiBSb290Q0EyMB4XDTA5MDUyOTA1MDAzoVoX
DTI5MDUyOTA1MDAzoVowXTELMaKGA1UEBhMCSlAxAjBjBgNVBAoTHFNfQ09NIFRy
dXN0IFN5c3RlbXMgQ08uLExURC4xKzAlBgNVBAStHINiY3VyaXR5IENvbW11bmlj
YXRpb24gUm9vdENBMjCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANAV
OVKxUrO6xVmCxF1SrjpDZYBLx/KWvNs2l9amZIyoXvDjChz335c9S672XewhtUGr
zbl+dp+++T42NKA7wfYxEUv0kz1XgMX5iZnK5atq1LXaQZAQwdbWQonCv/Q4EpVM
VAX3NuRFg3sUZdbcDE3R3n4MqzvEFb46VqZab3ZpUql6ucjrappdUtAtCms1FgkQ
hNBqyjoGADdH5H5XTz+L62e4iKrFvlnVspHEfbmwhRkGeC7bYRr6hfVKkaHnFtWO
ojnfilhwHyg/i/xAXmODPIMqGplrz95Zajv8bxbXH/1KEOtOghY6rCcMU/Gt1SSw
awNQwS08Ft1ENCcadfsCAwEAAaNCMEAwHQYDVR0OBBYEFAqFqXdlBZh8QIH4D5cs
OPEK7DzPMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3
DQEBChUA4IBAQBMOqNErLlFscEtfsgLcKlfZOoc7llsCLqJX2rKSpWeeo8HxdpF
coJxDjrSzG+ntKEju/Ykn8sX/oymzsLS28yN/HH8AynBbF0zX2S2ZTuJbxb2ePXc
okgfGT+Ok+vx+hufuzU7jBBJV1uXk3fs+BXziHV7Gp7yXT2g69ekuCkO2r1dcYmh8
t/2jioSgrGK+KwmHNPBqAbubKVY8/gA3zyNs8U6qtnRGEmyR7jTV7JqR50S+kDFy
1UkC9gLl9B/rfNmWVAn/7lr5mUf/NVoCqgTLiluHcSmRvaS0eg29mvVXIwAHIRc/
SjnRBuKlp7Y3gaVdjKozXoEofKd9J+sAro03
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIID3TCCAsWgAwIBAgIBADANBgkqhkiG9w0BAQsFADCBjzELMAKGA1UEBhMCMVVMx
EDAObgNVBAgTB0FyaXpnbmExEzARBgNVBAcTCINjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xMjAwBgNVBAMTKVN0YXJmaWVs
ZCBzSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5MDkwMTAwMDAw
MFoXDTM3MTIzMTIzNTk1OVowgY8xCzAJBgNVBAYTAiVtMRAwDgYDVQQGEwdBcm16
b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKExxTdGFyZml1bGQgVGvGj
aG5vbG9naWVzLCBjbmMuMTIwMAZDVQQDEylTdGFyZml1bGQgUm9vdCBDZXJ0aWZp
Y2F0ZSBDbXR0b3JpdHkgLSBMHjCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAL3twQP89o/8ArFvW59I2Z154qK3A2FWGMNHttfKPTUuiUP3oWmb3ooa/RMg
nLRJdzlpVv257IzdIvpy3Cdhl+72WoTsbhm5iSzchFvVdPtrX8WJpRBSiUZV9Lh1
HOZ/5FSuS/hVclCGfgXcVnrHigHdMWdSL5stPSksPNkN3mSwOxGXn/hbVNMYq/N

Hwtjuzqd+/x5AJhhdM8mgkBj87JyahkNmcrUDnXMN/uLicFZ8WJ/X7NfZTD4p7dN
dloedl40wOiWVpmKs/B/pM293DIxfJHP4F8R+GuqSVzRmZTRouNjWwl2tVZi4Uto
HZbUJtQIBFnQmA4O5t78w+wfkPECAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAO
BgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFHwMMh+n2TB/xH1oo2KooC6rB1snMA0G
CSqGSib3DQEBcWUAA4IBAQAARWfolTwNvlJk7mh+ChTnUdgWUXuEok21iXQnCoKjU
sHU48TRqneSfioYmUeYs0cYtbpUgSpIB7LiKZ3sx4mcujJUDJi5DnUox9g61DLu3
4jd/IroAow57UvtruzvE03IRTs2Q9GcHGcg8RnoNAX3FWOdt5oUwF5okxBDgBPfg
8n/Uqgr/Qh037ZTlZFkSIHc40zI+OIF1lnP6aI+xy84fxyz6nH7PfrHxBy22/L/K
pL/QlwVKvOoYKAKQvVR4CSFx09F9HdkWsKlhPdAKACL8x3vLCWRFCztAgfd9fDL1
mMpYjn0q7pBZc2T5NnReJaH1ZgUufzkVqSr7UluOhWn0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIID7zCCAtagAwIBAgIBADANBgkqhkiG9w0BAQsFADCbmDELMaKGA1UEBhMCVVMx
EDAObgNVBAGTB0FyaXpnbmExEzARBgNVBAcTClnjb3R0c2RhbGUxJTJhBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAAtIEcyMB4XDTA5
MDkwMTAwMDAwMFoXDTM3MTIzMTIzNTk1OVowgZgxGzAJBgNVBAYTAiVtMRAwDgYD
VQQIEwdBcm16b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKEExTdGFy
ZmllbGQgVGvjaG5vbG9naWVzLCBjb25hMRMwEQYDVQQDEzJTdGFyZmllbGQgU2Vy
dmljZXMGUm9vdCBDZXJ0aWZpY2F0ZSBDbXR0b3JpdHkgLSBMjCCASiWdQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBANUMOsQq+U7i9b4Zl1+OiFOxHz/Lz58gE20p
OsgPftZ3a3Y4Y9k2YKibXlwAgLIvWX/2h/klQ4bnaRtSmpDhcePYLQ1Ob/bISdm2
8xpWriu2dBTrz/sm4xq6HZYujtYlIHVv8loJNwU4PahHQUw2eeBGg6345AWH1K
Ts9DkTvnVtYAcMtS7nt9rjrnvDH5RfbCYM8TWQIrgMw0R9+53pBlbQLPLJGmpufe
hRhJfGZOozptqbXuNC66DQO4M99H67FrjSXZm86B0UVGMpZwh94CDklDhbZsc7tk
6mFBrMnUVN+HL8cisibMn1lUaJ/8viovxFUcdUBgF4UCVTmLfwUCAwEAAaNCMEAw
DwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFJxfAN+q
AdcwKziIorhtSpzyEZGDMA0GCSqGSib3DQEBcWUAA4IBAQBLLNqaEd2ndOxmFZyMI
bw5hyf2E3F/YNoHN2BtBLZ9g3ccaaNnRb0bhiCPPE95Dz+I0swSdHynVv/heyNXB
ve6SbzJ08pGCL72CQnqtKrcgfU28elUSwhXqvfdqlS5sdJ/PHLTyxQGjhdByPq1z
qwubdQxtRbeOlKyWN7Wg0I8VRw7j6IPdj/3vQQF3zCepYoUz8jcI73HPdwbyBkd
iEDPfuYd/x7H4c7/I9vG+o1VTqkC50cRRj70/b17KSa7qWFiNyi2LSr2EIZkyXCn
0q23KXB56jzaYyWf/Wi3MOxw+3WKt21gZ7IeyLnp2KhvAotnDU0mV3HaIPzBSICN
sSi6
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFujCCA6KgAwIBAgIJALtAHEP1Xk+wMA0GCSqGSib3DQEBBQUAMEUxCzAJBgNV
BAYTAkNIMRUwEwYDVQQKEwxTd2lzc1NpZ24gQUcxHzAdBgNVBAMTFIN3aXNzU2ln
biBhb2xkIENBIC0gRzIwHhcNMDYxMDI1MDgzMDM1WhcNMzYxMDI1MDgzMDM1WjBF
MQswCQYDVQQGEwJDSDEVMGMGA1UEChMMU3dpc3NTaWduIEFHMR8wHQYDVQQDEzT
d2lzc1NpZ24gR29sZCBdQSAAtIEcyMIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIIC
CgKCAgEAr+TufoskDhJuqVAtFkQ7kpJcyrhdhJJCEyq8ZVeCQD5XJM1QiyUqt2/8
76LQwB8CJEoTlo8jE+YoWACjR8cGp4QjK7u9lit/VcyLwVcfDmJID909Vopz2q5+
bbqBHH5CjCA12UNNhPqE21Is8w4ndwtrvxEvcnifLtg+5hg3Wipy+dpikJKVyh+c

6bM8K8vzARO/Ws/BtQpgvd21mWRTuKCWs2/iJneRjOBiEAKfNA+k1ZlZUd6+jbqE
emA8atufK+ze3gE/bk3lUIbLtK/tREDFylQm2lrfKjuvqblCqoOpd8FUrdVxyJd
MmqXl2MT28nbeTZ7hTpKxVKJ+STnnXepgv9VHKVxaSvRAiTysybUa9oEVeXBCsdt
MDeQKuSeFDNeFhdVxVu1yzSJkvGdJo+hB9TGsnhQ2wwMC3wLjEHXuednjIj3o02y
MszYF9rNt85mndT9Xv+9lz4pded+p2JYryU0pUHHpBwNUMoDAw8IWWh+Vc3hiv69y
FGkOpeUDDniOJihC8AcLYiAQZzlG+qkDzAQ4embvII01jEpWjpEA/I5cgt6IoMPi
aG59je883WX0XaxR7ySArqpWl2/5rX3aYT+YdzyLkbYcjCbaZaIJbCHiVOO5yKxM
gl93e2CaHt+28kgeDrpOVG2Y4OGiGqJ3UM/EY5LsRxmd6+ZrzsECAwEAAaOBRDCB
qTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWyV7
lqRIUX64OfPAeGZe6Dm8O4wHwYDVR0jBBgwFoAUWyV7lqRIUX64OfPAeGZe6Dm
8O4wRgYDVR0gBD8wPTA7BgIghXQBWQECAQEwLjAsBggrBgEFBQcCARYgaHR0cDov
L3JlcG9zaXRvcnkuc3dpc3NzaWduLmNvbS8wDQYJKoZIhvcNAQEFBQADggIBACe6
45R88a7A3hfm5djV9VSwg/S7zV4Fe0+fdWavPOhWfvxyeDgD2StiGwC5+OlgzczO
UYrHUDFu4Up+GC9pWbY9ZIEr44OE5iKHjn3g7gKZYbge9LgriBIWhMIxkziWMaa5
O1M/wySTVltpkuzFwbs4AOPsF6m43Md8AYOfMke6UiI0HTJ6CVanfCU2qT1L2sCC
bwq7EsiHSycR+R4tx5M/nttfJmtS2S6K8RTGRI0Vqbe/vd6mGu6uLftIdxf+u+yv
GPUqUfA5hJeVbG4bwyvEdGB5JbAKJ9/fXtI5z0V9QkvfsywexcZdylU6oJxpmo/a
77KwPJ+HbBIRZXAVUjEaJM9vMSNQH4xPjyPDdEFjHFWoFN0+4FFQz/EbMFYOkRCC
hdiDyyJkvC24JdVUorgG6q2SpCSgwYa1ShNqR88uC1aVVMvOmttqtKay20EIhid3
92qgQmwLOM7XdVAyKsLfKzAiSNDVQTglXaTpXZ/GIHXQRf0wl0OPkKsKx4ZzYEpp
Ld6leNcG2mqeSz53OiATlGHQv2ieY2BrNU0LbbqhPcCT4H8js1WtciVORvnSFu+w
ZMEBnunKoGqYDs/YYPivSbjkQuE4NRb0yG5P94FW6LqjviOvr1vA+ACozB2+htt
Qc8Bsem4yWb02ybzOqR08kkkW8mw0FfB+j564ZfJ

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFkzCCA3ugAwIBAgIUQ/oMX04bgBhE79G0TzUfRPSA7cswDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCQ0gxFTATBgNVBAoTDFN3aXNzU2lnbiBBRzErMCKGA1UE
AxMiU3dpc3NTaWduIFJTSBUTFMgUm9vdCBDQSAyMDIyIC0gMTAeFw0yMjA2MDgx
MTA4MjJaFw00NzA2MDgxMTA4MjJaMFExCzAJBgNVBAYTAkNIMRUwEwYDVRQQKEwxT
d2lzc1NpZ24gQUcxKzApBgNVBAMTIlN3aXNzU2lnbiBSU0EgVExTIFJvb3QgQ0Eg
MjAyMiAtIDEwgglMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDLKmjiC8NX
vDVjvHCIO/OMPE5Xlm7DTjak9gLKHQquuN6orx122ro10JFwB9+zBvKK8i5VUXu7
LCTLf5ImgKO0lPaCoaTo+nUdWfMHamFk4saMla+ju45vVs9xzF6BYQ1t8qsCLqSX
5XH8irCRIFucdFJtrhUnWXjyCcplDn/L9Ovn3KlMd/YrFgSVrpxpT8q2kFC5zyE
EPThPYxr4iuRR1VPuFa+Rd4iU1U1OKNlfGUEGjw5NBuBwQCMBauTLE5tze0USJlt
/m2n+IdreXXhvhCxpqhAWVTXz8TQm0SzOGlkjIHRI36qOTw7D59Ke4LKa2/KIj4x
0LDQKhySio/YGZxH5D4MucLNvkEM+KRHBdvBFzA4OmnczcNpI/2aDwLOEGrOyvi5
KaM2iYauC8BPY7kGWUleDsFpswrzd34unYyzJ5jSmY0lpx+Gs6ZUCdj8fV3oT4MM
0ZPIEuRU2j7yrTrePjxF8CgPBrnh25d7mUWe3f6VWQQvdT/TromZhqwUtKiE+shd
OxtYk8EXlFXIC+OCeYSf8wCENO7cMdWP8vpPlkwGqnj73mSiI80fPsWMvDdUDrta
clXvyFu1cvh43zcgTFeRc5JzrBh3Q4IgaezprClG5QtO+DdziZaKHG29777YtvTK
wP1H8K4LWCDfYb02rpeNUIMmJcn3nTsPBQIDAQABo2MwYTAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAfBgNVHSMEGDAWgBRvjmKLk0Ow4UD2p8P98Q+4
DxU4pTAdBgNVHQ4EFgQUb45ii5NDsOFA9qfD/fEPuA8VOKUwDQYJKoZIhvcNAQEL
BQADggIBAKwsKUF9+lz1GpUYvyypiqkkVHX1uECry6gkUSsYP2OprphWKwVDIqO3
10aewCoSPY6WlkDfDDOLazeROpW7OSltWjSipQLBwJNGD77+3v1dj2b9l4wBlgz

Hqp41eZUBDqyggmNzhYzWUUo8aWjlw5DI/0LIICQ/+Mmz7hkkeUFjxOgdg3XNwwQ
iJb0Pr6VvfHDffCjw3lHC1ySFWPtUnWK50Zpy1FVCypM9fJkT6lc/2cyjlUtMoIc
gC9qkfjLvH4YoiaoLqNTKIftV+Vlek4ASltOU8liNr3CjlvzG4ngRhZi0Rjn9UM
ZfQpZX+RLOV/fuiJz48gy20HQhFRJjKKLjpHE7iNvUcNCfAWpO2Whi4Z2L6MOuhF
LhG6rlrnub+xzl/goP+4s9GFe3lmozmlO2bYQL7Pt2eLSMkZJVX8vY3PXtpOpvJp
zv1/THfQwUY1mFwjmwJFQ5Ra3bxHrSL+ul4vkSkphnsh3m5kt8sNjzdbowhq6/Td
Ao9QAwKxuDdollDruF/UKIqlIgyKhPBZLtU30WHlQnNYKoH3dtvi4k0NX/a3vgW0
rk4N3hY9A4GzJl5LuEsAz/+MF7psYC0nhzck5npgL7XTgwSqT0N1osGDsieYK7EO
gLrAhV5Cud+xYJHT6xh+cHiudoO+cVrQkOPKwRYlZ0rwtnu64ZzZ

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDwzCCAaquAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50ZXIxJTAjBgNVBAMMHFQtVGVsZVNl
YyBHbG9iYWxSb290IENsYXNzIDlwHhcNMDgxMDAxMTA0MDE0WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50
ZXIxJTAjBgNVBAMMHFQtVGVsZVNlYyBHbG9iYWxSb290IENsYXNzIDlwggEiMA0G
CSqGSib3DQEBAQUAA4IBDwAwggEKAoIBAQCqX9obX+hzkeXaXPSi5kfl82hVYAud
AqSzm1nzHoqyNK38DcLZSBnuaY/JIPwhqgcZ7bBcrGXHX+0CfHt8LRvWurmAwhiC
FoT6ZrAlxlQjgeTNUk/9k9uN0goOA/FvudocP05l03Sx5iRUKrERLMjfTIH6VJi
1hKTXrcxlkIF+3anHqP1wvzpesVsqXFP6st4vGCvx9702cu+fjOlbpSD8DT6Iavq
jnKgP6TeMFvvhk1qlVtDRKgQFRzLA VfFmPHmBiiRqiDFt1MmUUOyCxGVWOHAD3bZ
wI18gfNycJ5v/hqO2V81xrJvNH+SE/iWjnx2J14np+GPgNeGYtEotXHA gMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBS/
WSA2AHmgoCJrjNXyYdK4LMuCSjANBgkqhkiG9w0BAQsFAAOCAQEAMQoiYQsfdOhy
NsZt+U2e+iKo4YFWz827n+qrkRk4r6p8FU3ztqONpfSO9kSpp+ghla0+AGIWiPAC
uvxhl+YzmsB6azZie60EI4RYZeLbK4rnJVM3YINfvNoBYimipidx5joifsFvHZVw
IEoHNN/q/xWA5brXethbdXwFeilHfkCoMRN3zUA7tFFHei4R40cR3p1m0IvVVGb6
g1XqfMIpiRvpb7PO4gWEyS8+eIVibslfwXhjdFjASBgMmTnrpMwatXlajRWc2BQN
9noHV8cigwUtPjJlJ0Ys6lDfMjIq2SPDqO/nBudMNva0Bkuqjzx+zOAduTNrRIP
BSeOE6Fuwg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDwzCCAaquAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50ZXIxJTAjBgNVBAMMHFQtVGVsZVNl
YyBHbG9iYWxSb290IENsYXNzIDMwHhcNMDgxMDAxMTA0MDE0WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50
ZXIxJTAjBgNVBAMMHFQtVGVsZVNlYyBHbG9iYWxSb290IENsYXNzIDMwggEiMA0G
CSqGSib3DQEBAQUAA4IBDwAwggEKAoIBAQC9dZPwYiJvJK7genasfb3ZJNW4t/zN
8ELg63iIVl6bmlQdTQyK9tPPcPRStdiTBONGhnFBSivwKixVA9ZIw+A5OO3yXDw/
RLyTPWGrTs0NvvAgJ1gORH8EGoel15YUNpDQSXuhdfsaa3Ox+M6pCSzyU9XDFES4

hqX2iys52qMzVnN6chr3IhUciJFrF2blw2qAsCTz34ZFiP0Zf3WHHx+xGwpzJFu5
ZeAsVMhg02YXP+HMVDNzkQI6pn97djmIH5a2OK61yJN0HZ65tOVgnS9W0eDrXltM
EnAMbEQgqxHY9Bn20pxSN+f6tsIxO0rUFJmtxxr1XV/6B7h8DR/Wgx6zAgMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBS1
A/d2O2GCahKqGFPrAyGUv/7OyjANBgkqhkiG9w0BAQsFAAOCAQEAVj3vlnW92nOy
WL6ukK2YJ5f+AbGwUgC4TeQbIXQbfsDuXmkqJa9c1h3a0nnJ85cp4IaH3gRZD/FZ
1GSFS5mvJQQeyUapl96Cshtwn5z2r3Ex3XsFpSzTucPH9sry9uetuUg/vBa3wW30
6gmV7PO15wWeph6KU1HWk4HMDJP2udqmJQV0eVp+QD6CSyYRMG7hP0HHRwA11fXT
91Q+gT3aSWqas+8QPeb9HIIkflZM8BMZLZGOMivgkeGj5asuRrDFR6fUNOuIml
e9eiPZaGzPImNC1qkp2aGtAw4l1OBLBfiyB+d8E9IYLRRpo7PHi4b6HQDWSieB4p
TpPDpFQUWw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIeYzCCA0ugAwIBAgIBATANBgkqhkiG9w0BAQsFADCB0jELMAkGA1UEBhMCVFIX
GDAWBgNVBAcTD0dlYnplIC0gS29jYWVsaTFCMEAGA1UEChM5VHVya2l5ZSBcWxp
bXNlbCB2ZSBuZwub2xvamlrIEFyYXN0aXJtYSBLdXJ1bXUgLSBUUVUJJVEFLMS0w
KwYDVQQLZyRLYW11IFNlcnRpZmlrYXN5b24gTWVya2V6aSAAtIEthbXUgU00xNjA0
BgNVBAMTLVRVQklUQUsgS2FtdSBTTSBTU0wgS29rIFNlcnRpZmlrYXNpIC0gU3Vy
dW0gMTAeFw0xMzExMjUwODI1NTVaFw00MzEwMjUwODI1NTVaMIHSMQswCQYDVQQG
EwJUUjEYMBYGA1UEBxMPR2ViemUgLSBLb2NhZWxpMUlwQAYDVQQKEzlUdXJraXll
IEJpbGltc2VsIHZlIFRla25vbG9qaWsgQXJhc3Rpcm1hIEt1cnVtdSAAtIFRVQklU
QUxSLTAeBgNVBAAsTjEethbXUgU2VydGlmaWthc3lvbiBNZXJrZXppIC0gS2FtdSBT
TTE2MDQGA1UEAxMtVFVCSVRBSyBLYW11IFNlcnRpZmlrYXN5b24gTWVya2V6aSAAtIEthbXUgU00xNjA0
LSBTdXJ1bSAXMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3UwM6q7
a9OZLBI3hNmNe5eA027n/5tQlT6QIVZC1xl8JoSNkvoBHToP4mQ4t4y86lj5iySr
LqP1N+RAjhgleYN1Hzv/bKjFxlB4tO2KRKOrbEz8HdDc72i9z+SqzvBV96I01INr
N3wcwv61A+xXzry0tcXtAA9TNypN9E8Mg/uGz8v+jE69h/mniyFXnHrfA2eJLJ2X
YacQuFWQfw4tJzh03+f92k4S400VIgLI4OD8D62K18IUUMw7D8oWgITQUVbDjlZ/
iSizL+aFCr2lqBs23tPcLG07xxO9WSMs5uWk99g7LeqQQESolbuT1dCANLZGeA4f
AJNG4e7p+exPFwIDAQAB0IwQDAdBgNVHQ4EFgQUZT/HiobGPN08VFw1+DrtUgxH
V8gwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBACo/4fEYjq7hmFxLXs9rHmoJ0iKpEsdeV31zVmSAhHqT5Am5EM2fKifh
AHe+SMg1qIGf5LgSyX8OsNjLN13qudULXjS99HMPw+0mFZx+CFOKWI3QSyjfwBpf
IPP54+M638yclNhOT8NrF7f3cuitZjO1JVOr4PhMqZ398g26rmZqsZr+ZO7rqu4
lzwDGrpDxp5RXI4s6ehl2Re37AIVNMh+3yC1SVUZPVIqUNivGTDj5UDrDYyU7c
8jEyVupk+eq1nRZmQnLzf9OxMUP8pI4X8W0jq5Rm+K37DwhuJi1/FwcJsoz7UMCf
lo3Ptv0AnVoUmr8CRPXbwp8iXqIPoeM=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFjTCCA3WgAwIBAgIQQAe0jMIAAAAAAAAAAATzyxjANBgkqhkiG9w0BAQwFADBQ
MQswCQYDVQQGEwJUVzESMBAGA1UEChMJVEFJV0FOLUNBMRAwDgYDVQQLEwdSb290
IENBMRSwGQYDVQQDEwJUV0NBIEZlcnR5b3QgQ0EwHhcNMjUyMDY1NDI5
WhcNNDCxMTIyMTU1OTU5WjBQMQUswCQYDVQQGEwJUVzESMBAGA1UEChMJVEFJV0FO
LUNBMRAwDgYDVQQLEwdSb290IENBMRSwGQYDVQQDEwJUV0NBIEZlcnR5b3Qg

Q0EwggliMA0GCSqGSib3DQEBAQUAA4ICDwAwggIKAoICAQDG+Moe2Qkgfh1sTs6P
40czRjZHyWmqOlt47nDSkvgEs1JSHWdyKKHfi12VCv7qze33Kc7wb3+szT3vsxxF
avcokPFhV8UMxKNQXd7UtsZyoC5dc4pztKFluwCY8xEMCDa6pFbVuYdHNWdZsc/
34bKS1PE2Y2yHer43CdTo0fhYcx9tbD47nORxc5zb87uEB8aBs/pJ2DFTxnk684i
JkXXYJndzk834H/nY62wuFm40AZoNWDtNq5xQwTxaWV4fPMf88oon1oglWa0zbfu
j3ikRRjpJi+NmykosaS3Om251Bw4ckVYsV7r8Cibt4LK/c/WMw+f+5eesRycnupf
Xtuq3VTpMCEobY5583WSjCb+3MX2w7DfRfIDo7YDKPYIMKoNM+HvnKkHIuNZW0CP
2oi3aQiotyMuRAIZN1vH4xfyIutuOVLf3lSnmMILIJXcRolftBL5hSmO68gnFSDA
S9TMfAxsNAwmmyYxpjyn9tnQS6Jk/zuZQXLB4HCX8SS7K8R0IrGsayIyJNN4KsDA
o/s/xUgXJP+92ZuJF2A09rZXIx4kmyA+upwMu+8Ff+iDhcK2wZSA3M2Cw1a/XDBzC
kHDXShi8fgGwsOsVhKQGzaRP6AzRwyAQ4VRlnrZR0Bp2a0JaWHY06rc3Ga4udfmW
5cFZ95RXKSWNOkyrTZpB0F8mAwIDAQABo2MwYTAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBSdhWEUfMfib5do5E83QOGt4A1WNzAd
BgNVHQ4EFgQUYvVhFHzBYm+XaORPN0DhreANVjcwDQYJKoZIhvcNAQEMBQAdggIB
AGSPesRiDrWiZLjHhg6hShbNcAu3p4ULs3a2D6f/CIsLJc+o1IN1KriWiLb73y0t
tGltITVX1olNc79pj3CjYcy2x6a4CD4bLubIp1dhDGaLlrdqHXKGnK/nZVekZn
68xDiBaiA9a5F/gZbG0jAn/xX9AKKSM70aoK7akXJlQKTcKlTfjF/biBzysseKNn
TKkHmvPfXvt89YnNdJdHEGoHK4Fa0o635yDRIG4kqIQnoVesqlVYL9zZyvp0BJ7t
RCT5dEA7IzOrg1oYJkK2bVS1FmAwBLGg+LhBoF1JsdlJBTrq/p1hvlbZv97Tujqx
f36SNI7JAG7cmL3c7IAFrQI932XtCwP39xaEBDG6k5TY8hL4iuO/Qq+n1M0RFxbI
Qh0UqEL20kCGoE8jypZfVvMAGzbdVAaYBlGX+bgUJurSkquLvWL69J1bY73NxW0Qz
8ppy6rBePm6pUlvscG21h483XjyMnM7k8M4MZ0HMzvaAq07MTFb1wWFZk7Q+ptq4
NxKfKjLji7gh7MMrZQzvlt6IKtTm1/r++FHvpw+PoP7UV31aPcuIYXcv/Fa4nzX
xeSDwWrruoBa3lwtcHb4yOWHh8qgnaHllhInD0Q9HWzq1MKLL295q39QpsQZp6F6
t5b5wR9iWqJDB0BeJas7a5wFsWqynKKTbDPAYsDP27X
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFQTCCAymgAwIBAgICDL4wDQYJKoZIhvcNAQELBQAUTELMAkGA1UEBhMCVFcx
EjAQBgNVBAoTCVRBSVdBt1DQTEQMA4GA1UECXMHUm9vdCBDQTEcMBoGA1UEAxMT
VFdDQSBHbG9iYWwUm9vdCBDQTAeFw0xMjA2MjcwNjI4MzNaFw0zMDEyMzExNTU5
NTlaMFExCzAJBgNVBAYTAiRlRMRiEAYDVQQKEwluQWUuXQU4tQ0ExEDAOBgNVBASt
B1Jvb3QgQ0ExHDAaBgNVBAMTE1RXQ0EgR2xvYmFsIFJvb3QgQ0EwggliMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCwBdvI64zEbooh745NnHEKH1Jw7W2CnJfF
10xORUnLQEK1EjRsGcJ0pDFfhQKX7EMzClPSnIyOt7h52yvVavKOZsTuKwEHktSz
0ALfUPZVr2YOy+BHYC8rMjk1Ujoog/h7FsYYuGLWRyWRzvAZEk2tY/XTP3VfKfCh
MBwqoJimFb3u/Rk28OKRQ4/6ytYQJ0IM793B8YVwm8rqqFpD/G2Gb3PpN0Wp8DbH
zlh1HrtsBv+baz4X7GGqcXzGHaL3SekVtTzWoWH1EfcFbx39Eb7QMAfCKbAJTibc
46KokWofwPFfiFzlmLhxpRUZyXx1EcxdwE8tmx2RRP1WKKD+u4ZqyPpcC1jcxkt2
yKsi2XMPpfRaAok/T54igu6idFMqPVMnaR1sjjIsZAAMy2E2TqNGtz99sy2sbZCi
laLOz9qC5wc0GZbpuCGqKX6mOL6OUhZnkfs8O1CWfe1tQHRvMq2uYiN2DLgbYP
oA/pyJV/v1WRBXRppRXAb94JlAGD1zQzbEcl8LibZ9WYkTunhHiVJqRaCPgrdLQA
BDzfuBSO6N+pjWxnkjMdwLfs7JLIvgm/LCkFbwJrnu+8vyq8W8BQj0FwcYeyTbcE
qYSjMq+u7msXi7Kx/mzhklyIqJdlzshNy/MGz19qCkKxHh53L46g5pIOBvwFItIm
4TFRfTLcDwIDAQABoyMwITAObgNVHQ8BAf8EBAMCAQYwDwYDVROTAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAgEAXzSBdu+WHdXltdkCY4QWwa6gcFGn90xHNcgl
1yg9iXHZqjNB6hQbbCEAwGxCGX6faVsgQt+i0trEfJdLjbDorMjupWkEmQqSpqsn

z6fLHgIwN0GMZt9Ba9aDAEH9L1r3ULRn0SyocddDypwnJJGDSA3PzfdUga/sf+Rn

27iQ7t0l

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFszCCA5ugAwIBAgIQIZxULEj27HF3+k7ow3BXlZANBgkqhkiG9w0BAQwFADBj
MQswCQYDVQQGEwJERTEnMCUGA1UECgweRGV1dHNjaGUgVGVsZWtvbSBTZWN1cmI0
eSBHbWJIMSSwKQYDVQQDDCJUZWxla29tIFNIY3VyaXR5IFRMUyBSU0EgUm9vdCAy
MDIzMB4XDTIzMDMyODEyMTY0NV0XDTQ4MDMyNzIzNTk1OVowYzELMAkGA1UEBhMC
REUxJzAlBgNVBAoMHkRldXRzY2hlIFRlbgVrb20gU2VjdXJpdHkgR21iSDErMCKG
A1UEAwwiVGVsZWtvbSBTZWN1cmI0eSBUTFMgUINBIFJvb3QgMjAyMzCCAiIwDQYJ
KoZihvcNAQEBBQADggIPADCCAgCggIBAO01oYGA88tKaVvC+1GDrib94W7zgRJ9
cUD/h3VCKSHtgVIs3xLBGYSJwb3FKNXVS2xE1kzbB5ZKVXrKNoIENqil/Cf2SFHV
cp6R+SPWcHu79ZvB7JPPGepIfohwoHP89v+1VmLhc2o0mD6CuKyVU/QBoCcHcqMA
U6DksquDOFcZJSfvgkdmOGjup5czQRxUX11eKvzWarE4GC+j4NSuHUaQTXtvPM6
Y+mpFEXX5ILRbtLevOP1Czvm4MS9Q2QTps70mDdsipWol8hHD/BeElvnHRz+sTug
BTNoBUGCwQMrAcjn02r6LX2zWtEtefdi+zqJbQAildNsLGyMcEWzv/9FIS3R/qy
8XDe24tsNlikfLMR0cN3f1+2JeANxdKz+bi4d9s3cXFH42AYTyS2dTd4uaNir73J
co4vzLuu2+QVUhhHM/tqty1LkCiCc/4YizWN26cEar7qwU02OxY2kTLvtkJkUPg
8qKrBC7m8kwOFjQgrIfBLX7JZkcXFBGk8/ehJImr2BrIoVyxo/eMbcgByU/J7MT8
rFEz0ciD0cmfHdRHNCk+y7AO+oMLKFjIKdw/fKifybYKu6boRhYPluV75Gp6SG12
mAWl3G0eQh5C2hrGUve1g8Aae3g1LDj1H/1Joy7SWWO/gLCMk3PLNaaZISJhZQN
g+Y+TS/qanIA7AgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUtqeX
gj10hZv3PJ+TmPV5dVKMbUcwDwYDVR0TAAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBS2
p5eCPXSfM/c8n5OalXl1UoxRzANBgkqhkiG9w0BAQwFAAOCAGEAqMxhpr51nhVQ
pGv7qHBFfLp+sVr8WyP6Cnf4mHGCDG3gXkaqk/QeoMPhk9tLrbKmXauw1GLLXrtm
9S3ul0A8Yute1hTWjOKWi0FpkzXmuZlrYrShF2Y0pmtjxrlO8iLpWA1WQdH6DErw
M807u20hOq6OcrXDSvvpfeWxm4bu4uB9tPcy/SKE8YXJN3nptT+/XOR0so8RYgDd
GGah2XsjX/GO1WfoVNpbOms2b/mBsTNHM3dA+VKq3dSDz4V4mZqTuXNnQkYRIer+
CqkbGmVps4+uFrb2S1ayLfmlyOw7YqPta9BO1UAJpB+Y1zqlklkg5LB9zVtzaL1t
xKITDmcZuI1CfmwMmm6gJC3VRRvcxAIU/oVbZZfKTPBQCHpCNfnqwmBU+AGuHrS+
w6jv/naaoqYfRvaE7fzbzsQCzndILIyy7MMAo+wsVRjBfhnu4S/yrYObnqsZ38aK
L4x35bcF7DvB7L6Gs4a8wPfc5+pbrLmTWGS9DiP7bY+A4A7l3j941Y/8+LN+l
X273CXE2whJdV/LItM3z7gLfEdxquVeEHVlNjM7IDiPCtyaaEBRx/pOyiriA8A4Q
ntOoUAw3gi/q4Iqd4Sw5/7W0cwDk90imc6y/st53BIe0o82bNSQ3+pCTE4FCxpgm
dTdmQRCSu/WU48IxK63nI1bMNSWSs1A=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFODCCAYCgAwIBAgIRAJW+FqD3LkbxezmCcvqLzZYwDQYJKoZihvcNAQEFBQA
wNzEUMBIGA1UECgwlVGVSaWFTb25lcmExHZAAdBgNVBAMMFIRlbgGhU29uZXJhIFJv
b3QgQ0EgdjEwHhcNMDCxMDE4MTIwMDUwWWhcNMZlXMDUwMDUwWjA3MRQwEgYD
VQQKDAUZWxpYVNVbmVYYTEfMB0GA1UEAwwVVGVSaWFTb25lcmEgUm9vdCBDBQSB2
MTCCAiIwDQYJKoZihvcNAQEBBQADggIPADCCAgCggIBAMK+6yfwIaPzaSZVfp3F
VRaRXP3vIb9TgHot0pGMYzHw7CTww6XScnwQbfQ3t+XmfHnqjLWCi65ltqwA3GV1
7CpNX8GH9SBIK4GoRz6JI5UwFpB/6FcHSOcZrr9FZ7E3GwYq/t75rH2D+1665I+X

Z75Ljo1kB1c4VWk0Nj0TSO9P4tNmHqTPGrdeNjPUtAa9GAH9d4RQAEX1jF3oI7x+
/jXh7VB7qTCNGdMJjmhnx88lxhTuylixcepsHHltTbLaC0H2kD7OriUPEMPPCs
81Mt8Bz17Ww5OXOAFshSsCPN4D7c3TxHoLs1iuKYalu+5b9y7tL6pe0S7fyYgKkm
dtwoSxAgHNN/Fnct7W+A90m7UwW7XWjH1Mh1Fj+JWov3F0fUTPHSiXk+TT2YqGHe
Oh7S+F4D4MHJHizTjU3TITazN19jY5szFPAAtJmtTfImMMsJu7D0hADnJoWjiUIMu
sDor8zagrC/kb2HCUQk5PotTubtn2txTuXZZNp1D5SDgPTJghSJrt8czu90VL6R4
pgd7gUY2B1bdeTXHlSw7sKMXNeVzH7RcWe/a6hBle3rQf5+ztCo3O3CLm1u5K7fs
slESl1MpWtTwEhDcTwK7EpIvYtQ/aUN8Ddb8WHUBiJ1YFkveupD/RwGJBmr2X7KQ
arMCpgKIv7NHfirZ1fpoeDVNAgMBAAGjPzA9MA8GA1UdEwEB/wQFMAMBAf8wCwYD
VR0PBAQDAgEGMB0GA1UdDgQWBbTwj1k4ALP1j5qWDNXr+nuqF+gTejANBgkqhkiG
9w0BAQUFAAOCAgEAvuRcYk4k9AwI//DTDGjkk0kiP0Qnb7tt3oNmzqjMDfz1mgbl
dxSR651Be5kqhOX//CHBXfDkH1e3damhXwIm/9fH907eT/j3HEbAek9ALCI18Bmx
0GtnLLCo4MBANzX2hFxc469CeP6nyQ1Q6g2EdvZR74NTxnr/DlZJLo961gzmJ1Tj
TQpgcmLNdKqFwPb/ImWvtxBnmq0wROMVvMeJuScg/doAmAyYp4Db29iBT4xdwNBed
Y2gea+zDTY4EzAvXUYNR0PVG6pZDrlcQZlIrXSHX8f8MVRBE+LHIQ6e4B4N4cB7
Q4WQxYpYxmUKEffxiMPAdkgS94P+5KFdSpcc41teyWRyu5FrgZLAMzTsVlQ2jqI
OylDRl6XK1TOU2+NSueW+r9xDkKLfP0ooNBlytrEgUy7onOTjsrDNYmiLbAJM+7
vVvrdX3pCi6GMyx5dwlpYn8s3CQh3aP0yK7Qs69cwsGJirQmx1wHiRszYd2qReW
t88NkvuOGKmYSdGe/mBEciG5Ge3C9THxOUilkCR1VBatzvT4aRRkOfujuLpwQMcn
HL/EVlP6Y2XQ8xwOFvVrhlhNGNTkDY6lnVuR3HYkUD/GKvvZt5y11ubQ2egZixVx
SK236thZiNSQvxaz2emsWWFUYBy6ysHK4bkgTI86k4mloMy/0/Z1pHWWbVY=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFdDCCA1ygAwIBAgIPAWdfJ9b+euPkrL4JWwWeMA0GCSqGSIb3DQEBCwUAMEQx
CzAJBgNVBAYTAkZJMR0wGAYDVQQKDBFUFZwXpYSBGaW5sYW5kIE95ajEZMBcGA1UE
AwwQVGVSaWEgUm9vdCBDQSB2MjMjAeFw0xODExMjMxMTU1NTRaFw00MzExMjMxMTU1
NTRaMEQxCzAJBgNVBAYTAkZJMR0wGAYDVQQKDBFUFZwXpYSBGaW5sYW5kIE95ajEZ
MBcGA1UEAwwQVGVSaWEgUm9vdCBDQSB2MjMjCAAIwDQYJKoZIhvcNAQEBBQADggIP
ADCCAgoCggIBALLQPwe84nvQa5n44ndp586dpAO8gm2h/oFIH0wnrI4AuhZ76zBq
AMCzdGh+sq/H1WKzeJ9Qyow2RCRj0jbpDIX2Q3bVTKFgcmfiKDolyzG4OiIjNLh9
vVYiQJ3q9HsDrWj8soFPmNB06o3lfc1jw6P23pLCWBnglrvFxKk9pXSW/q/5iaq9
lRdU2HhE8Qx3FZLgmEKnpNaqIJLNwaCzlrI6hEKNfdWV5Nbb6WLEWLN5xYzTNTOD
n3WhUidhOPFZPY5Q4L15POdsv5e2QJlt5c0BE0312/UqeBAMN/mUWZFdUXyApT
7GPzmX3MaRKgwhfwAZ6/hLzRUssbkmbOpFPlob/E2wnW5olWK8jjfN7j/4nlNW4o
6GwLI1GpJQXrSPjdscr6bAhR77cYbETKJuFzxokGgeWKrLDiKca5JLNRBh0pUPC
TEPlcDaMtjNXepUugqD0XBCzYYP2AgWGLnwtbNwDRm41k9V6lS/eINhbfSQBGq6
WT0EBXWdN6IOLj3rwaRSg/7Qa9RmjtzG6RJOHSpXqhC8fF6CfaamyftufUXJ63R
DolUK5X6wK0dmBR4M0KGCqlztft0DbcbMBnEWg4cJ7faGND/igFuvGqHKI3t+ZI
pEYslOqodmJHixBTB0hXbOKSTbauBcvwUpej6w9GU7C7WB1K9vBykLVAgMBAAGj
YzBhMB8GA1UdIwQYMBaAFHKs5DN5qkWH9v2sHZ7Wxy+G2CQ5MB0GA1UdDgQWBbRy
rOQzeapFh/b9rB2e1scvhtgkOTAObgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUw
AwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAoDtZpwmUPjaE0n4vOaWWl/oRrfxn83EJ
8rKJhGdEr7nv7ZbsnGTbMjBvZ5qsfl+yqwE2foH65IRE0qw24GtixX1LDoJt0nZi
0f6X+J8wfBj5tFJ3gh1229MdqfDBmgC9bXXYfef6xzijnHDoRnkDry5023X4blMM
A8iZGok1GTzTyVR8qPAs5m4HeW9q4ebqkYJpCh3DflminmtGFZhb069GHWLIzoBS
SRE/yQQSwxN8PzuKlts8oB4KtItUsiRnDe+Cy748fdHif64W1lZYudogsYMVoe+K

TTJvQS8TUoKU1xrBeKJR3Stwbbca+few4GeXVtt8YVMJAygCQMez2P2ccGrGKMOF
6eLtGpOg3kuYooQ+BXcBlj37tCAPnHICehIv1aO6UXivKitEZU61/Qrowc15h2Er
3oBXRb9n8ZuRXqWk7FlIEA04x7D6w0RtBPV4UBySllva9bguulvP5fBqnUsvWHMt
Ty3EHD70sz+rFQ47GUGKpMFXEmZxTPpT41frYpUJnlTd0cl8Vzy9OK2YZLe4A5pT
VmBds9hCG1xLEooc6+t9xnppxyd/pPiL8uSUZodL6ZQHCRJ5irLrdATczvREWeAW
ysUsWNCe8e89ihmpQfTU2Zqf7N+cox9jQraVpli/owd8k+BsHMYeB2F326CjYSlKA
rBPuUBQemMc=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFpTCCA42gAwIBAgIUZPYOZXdhags7tOqFhLuxibhxkw8wDQYJKoZIhvcNAQEM
BQAwWjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dp
ZXMsIEluYy4xJDAiBgNVBAMMG1RydXN0QXNpYSBhbG9iYWwgUm9vdCBDQSBHMzAe
Fw0yMTA1MjAwMjEwMTlaFw00NjA1MTkwMjEwMTlaMFoxCzAJBgNVBAYTAkNOMSUw
IwYDVQQKDDBxUcnVzdEFzaWEgVGVSjaG5vbG9naWVzLCBjbmuMSQwIgYDVQQDDBtU
cnVzdEFzaWEgR2xvYmFsIFJvb3QgQ0EgRzMWggIiMA0GCSqGSIb3DQEBAQUAA4IC
DwAwggIKAoICAQDAMYJhkuSUGwoqZdC+BqmHO1ES6nBBruL7dOoKjbmzTNyPtxNS
T1QY4SxzlZHFZjtqz6xjbYdT8PfxObegQ2OwxANdV6nnRM7EoYNI9lA+sX4WuDqK
AtCWHwDNBSHvBm3dIZwZQ0WhxeiAysKtQGIXBsaqvPPW5vxQfmZCHzyLpnl5hkA1
nyDvP+uLRx+PjsXUjrYsyUQE49RDdT/VP68czH5GX6zfZBCK70bwkPAPLfSIC7Ep
qq+FqklYqL9joDiR5rPmd2jE+SoZhLsO4fWviejL1AgdB4SQXMeJNnKziyhWTXA
yB1GJ2Faj/IN03J5Zh6fFZAHLf3ti1ZwA0pJpN9pMRJpXX5cynoTi+jm9WAPzJMs
hH/x/Gr8m0ed262IPfN2dTPXS6Tli/n1Q1hPy8gDVI+lhXgEGvNz8teHHUGf59gX
zhqcD0r83ERoVGjiQTz+LISGNzzNPY+i2+f3VANfWdP3kXjHi3dqFuVJhZBFcnAv
kV34PmVACxmZySYgWmjBNb9Pp1Hx2BErW+Canig7CjoKH8GB5S7wprlppYiU5msT
f9FkPz2ccEblooV7WIQn3MSAPmeamseaMQ4w7OYXQJXZRe0Blqq/DPNL0WP3E1jA
uPP6Z92bfW1K/zJmTSU7/xxnD4UiWQWRkUF3gdCFTIcQcf+eQxuulXUtgQIDAQAB
o2MwYTABBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFEDk5Plj7zjKsK5Xf/Ih
MBY027ySMB0GA1UdDgQWBRA5OTyI+84yrCuV3/yITAWNnu8kjAOBgNVHQ8BAf8E
BAMCAQYwDQYJKoZIhvcNAQEMBQADggIBACY7UeFNOPMyGLS0XuFlXsSUT9SnYaP4
wM8zAQLpw6o1D/GUE3d3NZ4tVlFEbuHGLige/9rsR82XRBf34EzC4Xx8MnpmyFq2
XFNFV1pF1AWZLy4jVe5jaN/TG3inEpQGAHUNcoTpLrxaatXeL1nHo+zSh2bbt1S1
JKv0Q3jbSwTEb93mPmY+KfJLaHEih6D4sTNjduMNHXJEIU/HHzp/LgV6FL6qj6j
ITk1dImmasI5+njPtqzn59ZW/yOSLlALqbUHM/Q4X6RJpstlcHboCoWASzY9M/eV
VHUl2qzEc4Jl6VL1XP04lQJqaTDFHApXB64ipCz5xUG3uOyfT0gA+QEEVcys+Tix
xHWVBqB/0Y0n3bOppHKH/lmLmnp0Ft0WpWIp6zqW3lunaFnT63eROfjXy9mPX1on
AX1daBli2MjN9LdyR75bl87yRaKZk62Uy5P2EgmVtqvXO9A/EcswFi55gORngS1d
7XB4tmBZrOfdRWOPyN9yaFvqHbgB8X7754qz41SgOAngPN5C8sLtLpvzHzW2Ntjj
gKGLzZlkD8KqQ7HK9W+eQ42EVJmzbsASZthwEPEGNTNDqJwuuhQxzhB/Hibjj9LV
+Hfsm6vxL2PZQl/gZ4FkkfGXL/xuJvYz+NO1+MRiqzFRJQJ6+N1rZdVtTTDIZbpo
FGWsJwt0ivKH

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICVTCCAAdygAwIBAgIUtyNkuI6XY57GU4HBdk7LKnQV1tcwCgYIKoZIzj0EAwMw
WjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dpZXMs

MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQwFAAOCAGeAIZtqBSBdGBanEqT3
Rz/NyjuujsCCztXIJXgXbODgcMTWltnZ9r96nBO7U5WS/8+S4PPFJzVXqDuiGev4
iqME3mmL5Dw8veWv0BIb5Ylrc5tvJQJLkIKvQMKtuppgJFqBTQUYo+IzeXoLH5Pt
7DIK9RME7I10nYEKqG/odv6LTytpEoYKNDbdgptvT+Bz3Ul/KD7JO6NXBNi2T2Wp
2xIQaOHEibgGIOcberyxk2GaGUARtWqFVwHxtlotJnMnlvm5P1vQiJ3koP26TpUJ
g3933FEFlJ0gcXax7PqJtZwuhfG5WyRasQmr2soaB82G39tp27RIGAAtvKLEiUuJ
pQ7hRGU+isFqMB3iYPg6qocJQrmBktwliJiJ8Xw18WLK7nn4GS/+X/jbh87qqA8M
pugLoDzga5SYnH+tBuYc6kIQX+ImFTw3OffXvO645e8D7r0i+yiGNFjEWn9hongP
XvPKnbwbPKfILfanIhHKA9jnZwqKDss1jjQ52MjqjZ9k4DewbNfFj8GQYSbbJIwe
SsCI3zWQzj8C9GRh3sflB5XeMhg6j6JCQCTl1jNdfK7vsU1P1FeQNWrcrgSXSyK0
ly4wBOeY99sLAZDBHwo/+ML+TvrbmNzFrwFuHnYWa8G5z9nODmxkKuU4CkUpijy
323imttUQ/hHWKNddBWcwauwxzQ=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF2jCCA8KgAwIBAgIMBfcOhtpJ80Y1LrqyMA0GCSqGSIlb3DQEBCwUAMIGIMQsw
CQYDVQQGEwJVUzERMA8GA1UECAwISWxsaw5vaXMxEDAOBgNVBACMB0NoaWNhZ28x
ITAFBgNVBAoMGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjExMC8GA1UEAwwOVHJ1
c3R3YXZlIEdsb2JhbCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0xNzA4MjMx
OTM0MTJaFw00MjA4MjMxOTM0MTJaMIGIMQswCQYDVQQGEwJVUzERMA8GA1UECAwI
SWxsaw5vaXMxEDAOBgNVBACMB0NoaWNhZ28xITAFBgNVBAoMGFRydXN0d2F2ZSBi
b2xkaW5ncywgSW5jLjExMC8GA1UEAwwOVHJ1c3R3YXZlIEdsb2JhbCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIB
ALldUShLPDeS0YLOvR29zd24q88KPUFd5dyqCblXAJ7mY2Hf8g+CY66j96xz0Xzn
swuvCAAJWX/NKSqIk4cXGIDtiLK0thAfLdZfVaITXdHG6wZWiyj+rDKd/VzDBcdu
7oaJuogDnXIhhpCujwOl3J+IKMujkkkP7NAP4m1ET4BqstTnoApTAbqOl5F2brz8
1Ws25kCI1nsvXwXoLG0R8+eyvpJETNKXpP7ScoFDB5zpET71ixpZfr9oWN0EACyW
80OzfpGZdNmcc9kYvkHHNHnZ9GLCQ7mzJ7Aiy/k9UscwR7PJPrhq4ufogXBeQotP
JqX+OsIgrv4Fo7NDK0G2x2EOFYeUY+VM6AqFcJNykbmROPDMjWLBz7BegIIT11
RtzuzWniTY+HKE40Cz7PFNm73bZQmq131BnW2hqIyE4bJ3XYsgjxroMwuREOzYfw
hI0Vcnyh78zyiGG69Gm7DIwLdVcEuE4qFC49DxweMqZiNu5m4iK4BUBjECLzMx10
coos9TkpoNPnG4CELCU9402x/RpvumUHO1jsQkUm+9jaXLE9gCxInm943xZYkqc
BW89zubWR2OZxiRvchLrH+QtAuRcOi35hYQcRfO3gZPSEF9NUqjifLJS3tBEW1n
twiYTOURGa5CgNz7kAXU+FDKvuStx8KU1xad5hePrzb7AgMBAAGjQjBAMA8GA1Ud
EwEB/wQFMAMBAf8wHQYDVR0OBBYEFjngGWcNYtt2s9o9uFvo/ULSMQ6HMA4GA1Ud
DwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAGeAaHNw4rDT7TnsTGDZqRKGFx6W
00hUKDtkLSGm+J1WE2pIPU/HPinbbViDVD2HfSMF1OQc3Og4ZYbFdada2zUFvXfe
uyk3QAUHw5RSn8pk3fEbK9xGChACMf1KaA0HZJDmHvUqoi7PF35owgLEQzxPy0Q
IG/+4jSHg9bP5Rs1bdID4bANqKCqRieCNqcVtgimQlRXtpla4gt5kNdXELE1GYhB
aCXUNxeEFfsBctyV3lImIJgm4nb1J2/6ADtKYdkNy1GTKv0WBpanI5ojSP5RvbbE
sLFUzt5sQa0WZ37b/TjNuThOssFgy50X31ieemKyJo90LzvkWx3SD92YHJtZuSPT
MaCm/zjdzyBP6VhWomfD0faZmZ26NraAL4hHT4a/RDqA5Dccprql5gR0IRiR2Qe
qu5AvzSxnI904fKSTx+O856X3vOmeWqJcU9LJxdI/uz0UA9PSX3MReO9ekDFQdxh
VicGaeVYQYHTtgJoC86cnn+OjC/QueHYj6RS8fZMXZC+fc8Y+wmjHMMfRod6qh8
h6jCJ3zhM0EPz8/8AKAigJ5Kp28AsEFFtyLKAEjFQqKu3R3y4G5OBVixwJAWKqQ9
EEC+j2Jjg6mcgn0tAumDMHzLJ8n9HmYAsC7TIS+OMxZsm00QqAfWzJPP29FpHOTK
yeC2nOnOcXHebD8WpHk=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICYDCCAgewAwIBAgIMDWpfCD8oXD5Rld9dMAoGCCqGSM49BAMCMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAObGNVBAcTB0NoaWNhZ28xITAf
BgNVBAoTGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3
YXZlIEEdsb2JhbCBFQ0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM1MTBaFw00MjA4MjMxOTM1MTBaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAObGNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEEdsb2JhbCBF
Q0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTBZMBMGByqGSM49AgEGCCqG
SM49AwEHA0IABH77bOYj43MyCMpg5lOcuSNGLB4kFKA3TjASh3RqMyTpJcGOMoN
FWLGjgEqZ2q3zSRL0HB5DOSMcT9CTqmP62jQzBBMA8GA1UdEwEB/wQFMAMBaf8w
DwYDVR0PAQH/BAUDAwcGADAdBgNVHQ4EFgQUo0EGrJBt0UrrdaVKEJmzsaGLSvcw
CgYIKoZlZj0EAwIDRwAwRAIgB+ZU2g6gWrKuEZ+Hxbb/ad4lvvigtwjzRM4q3wgh
DDcCIC0mA6AFvWvR9lz4ZcyGbbOcNEhjhAnFjXca4syc4XR7

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICnTCCAisGAwIBAgIMCL2Fl2yZJ6SAaEc7MAoGCCqGSM49BAMDMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAObGNVBAcTB0NoaWNhZ28xITAf
BgNVBAoTGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3
YXZlIEEdsb2JhbCBFQ0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM2NDNaFw00MjA4MjMxOTM2NDNaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAObGNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEEdsb2JhbCBF
Q0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTB2MBAGByqGSM49AgEGBSuB
BAAiA2IABGvaDXU1CDFHBa5FmVXxERMuSvqQMSOjfoPTfygIOiYaOs+Xgh+AtycJ
j9GOMMQkmw6sWASr9zZ9lCokmwqKi6vr/TklZvFe/oyujUF5nQlgziip04pt89ZF
1PKYhDhloKNDMEEdwYDVR0TAQH/BAUwAwEB/zAPBgNVHQ8BAf8EBQMDBwYAMB0G
A1UdDgQWBBRVqYSJ0sEyyRjLbKYHTsjnnb6CkDAKBggqhkJOPQQDAwNnADBKAjA3
AZKXRRJ+oPM+rRk6ct30UJMDEr5E0k9BpIycnR+j9sKS50gU/k6bpZFXrsY3crsC
MGclCrEMXu6pY5Jv5ZAL/mYiykf9ijH3g/56vxC+GCsej/YpHpRZ744hN8tRmKVu
Sw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFszCCA5ugAwIBAgIUewLV4kBMkkaGFmddtLu7sms+/BMwDQYJKoZIhvcNAQEL
BQAwYTELMaKGA1UEBhMCVE4xNzA1BgNVBAoMLkFnZW5jZSB0YXRpb25hbGUzGUG
Q2VydGlmaWNhdGlubiBFbGVjdHJvbmldWUxGTAXBgNVBAMMEFR1blRydXN0IFJv
b3QgQ0EwHhcnMTkwNDI2MDg1NzU2WhcNNDQwNDI2MDg1NzU2WjBhMQswCQYDVQQG
EwJUTjE3MDUGA1UECgwuQWdlbmNlIE5hdGlubmFsZSBkZSBkZXJ0aWZpY2F0aW9u
IEVsZWNoZm9uX2F1ZTEZTEZMBcGA1UEAwwQVHVuVHJ1c3QgUm9vdCBDQTCcAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgocGgIBAMPN0/y9BFPdDCA61YguBUtB9YOCfvdZ
n56eY+hz2vYGqU8ftPkLHzmMmiDQfgbU7DTZhrx1W4eI8NLZ1KMKsmwb60ksPqxd

2JQDoOw05TDENX37Jk0bbjBU2PWARZw5rZzJJQRNmpA+TkBuimvNKWfGzC3gdOgF
VwpIUpp6Q9p+7FuaDmJ2/uqdHYVy7BG7NegfJ7/Boce7SBbdVtfMTqDhuazb1YMZ
GoXRIJfXyqNlC/M4+QKu3fZnz8k/9YosRxqZbwUN/dAdjH8KcwAWJeRTIAAHDOF
li/LQcKLEITDCSSJH7UP2dl3RxiSlGBcx5kDPP73lad9UKGAwqmDrViVWShbhlU
r8a83YFuB9tgYv7sEG7aaAH0gxupPqJbI9dkxt/con3YS7qC0lH4Zr8GRuR5KiY2
eY8fTpkdso8MDhz/yV3A/ZAQprE38806JG60hZC/gLkMjNWb1sjxVj8agIl6qelb
MlEsPvLfe/ZdeikZjuXIvTZxi11Mwh0/rViizz1wTaZQmCXcl/m4WEElcb9PuISg
jwBUffYRbVinljvrS5YnzWuioYasDXxU5mZMZl+QviGaAkYt5IPCgLnPSz7ofzwB
7I9ezX/SKEIBlYrilz0QIX32nRzFNKHsLA4KUiwSVXAkPcvCFDVDXSdOvsC9qnyW
5/yeYa1E0wCXAgMBAAGjYzBhMB0GA1UdDgQWBBQGmpsfU33x9aTI04Y+oXNZtPdE
ITAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFaAamx9TffH1pMjThj6hc1m0
90QhMA4GA1UdDwEB/wQEAWIBBjANBgkqhkiG9w0BAQsFAAOCAgEAqgVutt0Vyb+z
xiD2BkewhpMI0425yAA/l/VsJ4hxyXT968pk21vvHl26v9Hr7lXpuhbI87mP0zYu
QEkHdVneixCwSQXi/5E/S7fdAo74gShczNxr18UnH1YeA32gAm56Q6XKRm4t+v4
FstVEuTGfbvE7Pi1HE4+Z7/FXxtbUcoqgRYYdZ2vyJ/0Adqp2RT8JeNnYA/u8EH
22Wv5psymsNUk8QcCMNE+3tjEUPRahphanltkE8pjkcFwRJpadbGNjHh/PqAulxP
xOu3Mqz4dWEX1xAZufHSCe96Qp1bWgvUxpVOKs7/B9dPfHgGiPEZtdmYu65xxBzn
dFIY7wyJz4sfDZMAbBSSSFCp61cpABbjNhzi+L/wM9VBD8TMPN3pM0MBkRARHtG5
Xc0yGYuPjCB31yLEQtyEFpslbei0VXF/sHyZ03FJuc9SpAQ/3D2gu68zngowYI7b
nV2UqL1g52KAdoGDDIzMMEZJ4gzSqk/rYXHv5yJiqfdcZGyfFoxnNidF9Ql7v/YQ
CvGwjVRDjAS6oz/v4jXH+XTgbzRB0L9zZVcg+ZtnemZoJE6AZb0QmQZZ8mWvuMZH
u/2QeItBcy6vVR/cO5JyboTT0GFMDcx2V+IthSIVNg3rAZ3r2OvEhJn7wAzMMujj
d9qDRlueVSjAi1jTkD5OGwDxFa2DK5o=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWjCCA0KgAwIBAgIQT9Irr/VkyDOeTzRYZiNwYDANBgkqhkiG9w0BAQsFADBH
MQswCQYDVQQGEwJDTjJERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNVBAMMHFVDQSBF
eHRlbmRlZCBWYWxpZGF0aW9uIFJvb3QwHhcNMTUwMzEzMDAwMDAwWhcNMzgxMjMx
MDAwMDAwWjBHMqswCQYDVQQGEwJDTjJERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNV
BAMMHFVDQSBFeHRlbmRlZCBWYWxpZGF0aW9uIFJvb3QwggliMA0GCsQGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCpCQcoEwKwmeBkqh5DFnpzsZGgdT6o+uM4AHrsiWog
D4vFsJszA1qGxliG1cGFu0/GnEBNyr7uaZa4rYEWmnySBesFK5pI0Lh2PpbIILvS
sPGP2KxFRv+qZ2C0d35qHzwaUnoEPQc8hQ2E0B92CvdqFN9y4zR8V05WAT558aop
O2z6+I9tTcg1367r3CTueUWnhbYFiN6IXSV8l2RnCdM/WhUFhvMJHuxYMjMR83dk
sHYf5BA1FxvyDrFspCqjc/wJHx4yGVMR59mzLC52LqGj3n5qiAno8geK+LLNEOfi
c0CTuwjRP+H8C5SzJe98ptfRr5//lpr1kXuYc3fUfugH0mK1lTnj8/FtDw5lhIpj
VMWAuCeS31HJqcBCF3RiJ7XwzJE+oJKcmhUfzhTA8ykADNkUVkLo4KRel7sFsLz
KuZi2irbWWIQJUoqgQtHB0MGcIfS+pMRKXpITeuUx3BNr2fVUbGAIAEBtHoIppB/
TuDvB0GHR2qlXov7z1CymIsvw4m6WC31MJixNnI5fkkE/SmnTHnkBVfblLkWU41G
sx2VYvdWf6/wFlthWG82UBEL2KwrlRYaDh8IzTY0ZRBiZtWAXxQgXy0MoHgKaNYs
1+lvK9JKBZP8nm9rZ/+I8U6laUpSNwXqxhaN0sSZ0YIrO7o1dfdRUvJzyAfd5LQD
fwIDAQABo0IwQDAdBgNVHQ4EFgQU2XQ65DA9DfcS3H5aBZ8eNjr34RQwDwYDVR0T
AQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQELBQADggIBADaNa
l8xCFWQpN5smLNB7rhVpLGsaGvdftvkHTFng88nIua7Mui563MD1sC3AO6+fcAUR
ap8lTwEpcOPIDOHqWnzcSbvBHiqB9RZLcpHlOjG5qtr8nR/zXUACE/xOHAbKsxSQ
VBcZEhrxH9cMaVr2cXj0lH2RC47skFSOvG+hTKv8dGT9cZr4QQehzZHkPJrgmzI5

c6sq1WnIeJEmMX3ixzDx/BR4dxIOE/TdFpS/S2d7cFOFyrC78zhNLJA5wA3CXWvp
4uXViI3WLL+rG761KlCSF3Ru/H38j9CHJrAb+7lsq+KePRXBOy5nAliRn+/4Qh8s
t2j1da3Ptfb/EX3C8CSlrdP6oDyp+l3cpaDvRKs+1ujl5BOWF3sGPjLtx7dCvHaj
2GU4Kzg1USEODm8uNBNA4StnDG1KQTAYI1oyVZnJf+A83vbsea0rWBmirSwiGpWO
vpaQXUJXxPkUAzUrHC1RVwinOt4/5Mi0A3PCwSaAuwtCH60NryZy2sy+s6ODWA2C
xR9GUeOcGMyNm43sSet1UNWMKFfnKdDTajAshqx7qG+XH/RU+wBeq+yNuJkbL+vmx
cmtpzyKEC2IPrNkZAJSidjzULZrtBJ4tBmIQN1IchXIbJ+XMxjHsN+xjWZsLHXbM
fjKaiJUINIK73nZfdklJrX+9ZSCyycErdhh2n1ax

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRjCCAY6gAwIBAgIQXd+x2lqj7V2+WmUgZQOQ7zANBgkqhkiG9w0BAQsFADA9
MQswCQYDVQQGEwJDTjERMA8GA1UECgwIVW5pVHJ1c3QxGzAZBgNVBAMMElVDQSBH
bG9iYWwgRzIgUm9vdDAeFw0xNjAzMTEwMDAwMDBaFw00MDEyMzEwMDAwMDBaMD0x
CzAJBgNVBAYTAkNOMREwDwYDVQQKDAhVbmlUcnVzdDEbMBkGA1UEAwwSVUNBIEds
b2JhbCBHMiBSb290MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICGgKCAgEAxeYr
b3zvJgUno4Ek2m/LAfmZmqkywiKHUYUGRO8vDaBsGxUypK8FnFyIdK+35KYmToni9
kmugow2ifsqTs6bRjDXVdfkX9s9FxeV67HeToI8jrg4aA3++1NDtLnurRiNb/yzm
VHqUwCoV8MmNsHo7JOHXaOIxPAYzRrZUEaalLyJUKlgNAQLx+hVRZ2zA+te2G3/R
VogvGjqNO7uCEeBHANBSh6v7hn4PJGtAnTRnvI3HLYZveT6OqTwXS3+wmeOwcWDc
C/Vkw85DvG1xudLeJ1uK6NjGruFZfc8oLTW4lVYa8bJYS7cSN8h8s+1LgOGN+jlj
tm+3SJUIsUROhYw6AlQgL9+/V087OpAh18EmNVQg7Mc/R+zvWr9LesGtOxdQXGLY
D0tK3Cv6brxzks3sx1DoQZbXqX5t2Okdj4q1uViSukqSKwxW/YDrCPBeKW4bHAyy
j5OJrdu9o54hyokZ7N+1wxrrFv54NkzWbtA+FxyQF2smuvt6L78RHBgOLXMDj6DI
NaBa4kx1HXHHOThTeEDMg5PXCp6dW4+K5OXgSORIskfNTip1KnvyIvbJvgmRlld6
ilis7nCs+dw4wwcOxJORNanTrAmyPPZGpeRaOrvjUYG0lZFWJo8DA+DuAUlwznP
O6Q0ibd5Ei9Hxeepl2n8pndntd978XplFeRhVmUCAwEAaANCMEAwDgYDVROPAQH/
BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVROBBYEFiHEjMz15DD/pQwIX4wV
ZyF0Ad/fMA0GCSqGSIb3DQEBcwUAA4ICAQATZSL1jiutROTL/7lo5sOASD0Ee/oj
L3rtNtqzym325p7lX1iPyzcyochltq44PTUbPrw7tgTQvPlJ9Zv3hcU2tsu8+Mg5
1eRfB70VVJd0ysrtT7q6ZHafgbiERUIMjW+i67HM0cOU2kTC5uLqGOiiHycFutfl
1qnN3e92mI0ADs0b+gO3joBYDic/UvuUospeZcnWhNq5NXHzJsBPd+aBJ9J3O5oU
b3n09tDh05S60FdRvScFDcH9yBIw7m+NEssIndTUv4BFFJqIRNow6rSn4+7vW4LV
PtateJLbXDzz2K36uGt/xDYotgIVilQsnLAXc47QN6MUPJiVAAwpBVueSUMxX8fj
y88nZY41F7dXyDDZQVu5FLbowg+UMaeUmMxq67XhJ/UQqAHojhJi6IjMtX9GI8Cb
EGY4GjZGXyJoPd/JxhMnq1MGrKI8hgZlb7F+sSIEmqO6SWkoaY/X5V+tBIZkxbxqg
DMUIYs6Ao9Dz7GjevJPfH1t/gMRMTLGmhlrDO7gJzRSBuhjjVFc2/tsvfEehOjPI
+Vg7RE+xygKJBjYoaMVLuCaJu9YzL1DV/pqJuhgyklTGW+Cd+V7lDSKb9triyCGy
YiGqhkCyLmTTX8jjfhFnRR8F/uOi77Oos/N9j/gMHYIfLXC0uAE0djAA5SN4p1bX
UB+K+wb1whnw0A==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICjzCCAhWgAwIBAgIQXIuZxVqUxdJxVt7NiYDMJjAKBggqhkiOPQQDAzCBiDEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCK5ldyBKZXJzZXkxZDASBgNVBAcTC0plcnNI
eSBDaXR5MR4wHAYDVQQKEExVUaGUgVVNFUIRSvVNUIE5ldHdvcmxSLjAsBgNVBAMT

JVVTRVJUcnVzdCBFQ0MgQ2VydGlmaWNhdGlubiBBdXRob3JpdHkwHhcNMTAwMjAx
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCMVVMxEzARBgNVBAGT
Ck5ldyBKZXJzZXkxFDASBgNVBACTC0plcnNleSBDaXR5MR4wHAYDVQQKEhVUaGUg
VVFUURSVVNUIE5ldHdvcmsxLjAsBgNVBAMTJVVTRVJUcnVzdCBFQ0MgQ2VydGlma
aWNhdGlubiBBdXRob3JpdHkwdjAQBgcqhkiOPQIBBgUrgQQAIGNiAAQarFRaQflo
I+d61SRvU8Za2EurxtW20eZzca7dnNYMYf3boIkDuAUU7FfO7l0/4iGzzvfUinnng
o4N+LZfQYcTxmdwlkWORfzCjtHDix6EznPO/LlTsV+zfTJ/ijTjeXmjQjBAMB0G
A1UdDgQWBBQ64QmG1M8ZwpZ2dEl23OA1xmNjmiAObgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAKBgqhkiOPQQDAwNoADBlAjA2Z6EWCNzklwBBHU6+4WMB
zzuqQhFkoJ2UOQIReVx7Hfpkue4WQrO/islJxOzksU0CMQDpKmFHjFJKS04YcPbW
RNZu9YO6bVi9JNlWSOrvxKJGgYhqOkbRqZtNyWHa0V1Xahg=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF3jCCA8agAwIBAgIQAf1tMPyjylGoG7xkDjUDLTANBgkqhkiG9w0BAQwFADCB
iDELMAkGA1UEBhMCMVVMxEzARBgNVBAGTCk5ldyBKZXJzZXkxFDASBgNVBACTC0pl
cnNleSBDaXR5MR4wHAYDVQQKEhVUaGUgVVFUURSVVNUIE5ldHdvcmsxLjAsBgNV
BAMTJVVTRVJUcnVzdCBSU0EgQ2VydGlmaWNhdGlubiBBdXRob3JpdHkwHhcNMTAw
MjAxMDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCMVVMxEzARBgNV
BAGTCk5ldyBKZXJzZXkxFDASBgNVBACTC0plcnNleSBDaXR5MR4wHAYDVQQKEhVU
aGUgVVFUURSVVNUIE5ldHdvcmsxLjAsBgNVBAMTJVVTRVJUcnVzdCBSU0EgQ2Vy
dGlmaWNhdGlubiBBdXRob3JpdHkwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCAEmUXNg7D2wiz0KxXDXbtzSfTTK1Qg2HqiqBNCS1kCdzOiZ/MPans9s/B
3PHTsdZ7NygRK0faOca8Ohm0X6a9fZ2jY0K2dvKpOyuR+OJv0OwWlJAJPuLodMkY
tJHUYmTbf6MG8YgYapAiPLz+E/CHFHV25B+O1ORRxfFnRghRy4YUVD+8M/5+bJZ/
Fp0YvVGONaanZshyZ9shZrHUm3gDwFA66Mzw3LyeTP6vBZY1H1dat//O+T23LLb2
VN3I5xI6Ta5MirdcmrS3ID3KfyI0rn47aGYBROcBTkZTmzNg95S+UzeQc0PzMsNT
79uq/nROacdrrjGCT3sTHDN/hMq7MkztReJVni+49Vv4M0GkPGw/zJSZrM233bkf6
c0Plfg6lZrEpFDKEY1WJxA3Bk1QwGROs0303p+tdOmw1XNtB1xLaqUkL39iAigmT
Yo61Zs8liM2EuLE/pDkP2QKe6xJMIxzzawWpXhaDzLhn4ugTncxbgtNMs+1b/97l
c6wjOy0AvzVVdAlJ2EIYGn+SNuZRkg7zJn0cTRe8yexDJtC/QV9AqURE9JnnV4ee
UB9XVKg+/XRjL7FQZQnmWEIuQxpMtPAIR1n6BB6T1CZGSICBst6+eLf8ZxXhyVeE
Hg9j1uliutZfVS7qXMYoCAQlObgOK6nyTJccBz8NUvXt7y+CDwIDAQABo0IwQDAd
BgNVHQ4EFgQUU3m/WqorSs9UgOHYm8Cd8rIDZsswDgYDVROPAQH/BAQDAgEGMA8G
A1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAFeUfA3P9wF9QZlIDHPF
Up/L+M+ZBn8b2kMVn54CVVeWFPFSPCeHLCjtHzoBN6J2/FNQwISbxmtOuowhT6KO
VWKR82kV2LyI48SqC/3vqOILVSoGIG1VeCkZ7l8wXEskEVX/JJpuXior7gtNn3/3
ATiUFJVDWbn7YKnuHKsSjKCaXqeYalltiz8I+8jRRa8YFWSQEG9zKC7F4iRO/Fjs
8PRF/iKz6y+O0tFYQXBl2+odnKPi4w2r78NBc5xjeambx9spnFixdjQg3IM8WcR
iQycE0xyNN+81XHfqHd4blsjDwSXWxavVcStkNr/+XeTWYRUc+ZruwXtuhxkYze
Sf7dNXGfSeUHM9h4ya7b6NnJSF5t0dCy5oGzuCr+yDZ4XUmFF0sbmZgIn/f3gZ
XHlKYC6SQK5MnyosycdiyA5d9zZbyuAlJQG03RoHnHcAP9Dc1ew91Pq7P8yF1m9/
qS3fuQL39ZeatTXaw2ewh0qpKJ4jjv9cJ2vhsE/zB+4ALtRZh8tSQZXq9EfX7mRB
VXyNWQKV3WKdwrnuWih0hKWbt5DHDAff9Yk2dDLWKMgwsAvgnEzDHNb842m1R0aB
L6KCq9NjRHDEjf8tM7qtj3u1cIiuPhnPQCjY/MiQu12ZivVS5ljFH4gxQ+6IHdfG
jjxDah2nGN59PRbxYvnKkKj9
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDODCCAiCgAwIBAgIGIAYYFFnACMA0GCSqGSIb3DQEBBQUAMDsx CzAJBgNVBAYT
AIJPMREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECXMqY2VydFNJR04gUk9PVCBD
QTAEfw0wNjA3MDQxNzIwMDRaFw0zMTA3MDQxNzIwMDRaMDsx CzAJBgNVBAYTAIJP
MREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECXMqY2VydFNJR04gUk9PVCBDQTCC
ASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALczuX7IJUqOtdu0KBuqV5Do
0SLTZLrTk+jUrIzhQGpgV2hUhE28alQCBf/fm5oqrl0Hj0rDKH/v+yv6efHHrfAQ
UySQi2bJqlirr1qjAOm+ukbuW3N7LBeCgV5iLKECZbO9xSsAfsT8AzNXDe3i+s5d
RdY4zTW2ssHQnIFKquSyAVwdj1+ZxLGt24gh65AIgoDzMKND5pCCrIUoSe1b16kQ
OA7+j0xbm0bqQfWwCHTD0IgztnzXdN/chNFDDnU5oSVAKOp4yw4sLjmdjItuFhwv
JoIQ4uNlIAoEwF73XVv4EOLQunpL+943AAAaWyjj0pxzPjKHmKHJUS/X3qws08C
AwEAAANCMEEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAcYwHQYDVR0O
BBYEFOCMm9slSbPxflbWskKHC9BroNnkMA0GCSqGSIb3DQEBBQUAA4IBAQA+0hyJ
LjX8+HXD5n9liPRyTMks1zJO890ZeUe9jtbkw9QSSQTaxQGcu8J06Gh40CEyecY
MnQ8SG4Pn0vU9x7Tk4ZkVJdclDVVc/6IJMCopvDI5NOFIV2oHB5bc0hH88vLbwZ
44gx+FkagQnIl6Z0x2DEW8xXjrJ1/RsCCdtZb3KTafcxQdaIOL+Hsr0Wefmq5L6I
Jd1hJyMctTEHBDa0GpC9oHRxUiltvBTjD4au8as+x6AJzKNI0eDbZOeStc+vckNw
i/nDhDwTqn6Sm1dTk/pwwpEOMfmbZ13pljheX7NzTogVZ96edhBiIL5VaZVDADIN
9u6wWk5JRFryX0KD

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRzCCAy+gAwIBAgIJEQA0tk7GNi02MA0GCSqGSIb3DQEBCwUAMEEExCzAJBgNV
BAYTAIJPMRQwEgYDVQQKEwtDRVJU0lHTiBTQTEcMBoGA1UECXMqY2VydFNJR04g
Uk9PVCBDQSBHMjAeFw0xNzAyMDYwOTI3MzVaFw00MjAyMDYwOTI3MzVaMEEExCzAJ
BgNVBAYTAIJPMRQwEgYDVQQKEwtDRVJU0lHTiBTQTEcMBoGA1UECXMqY2VydFNJ
R04gUk9PVCBDQSBHMjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAMDF
dRmRfUR0dIf+DjuW3NgBFszuY5HnC2/OOwppGnzC46+CjobXXo9X69MhWf05N0Iw
vIdqtg+piNguLWkh59E3GE59kdUWX2tbAMI5Qw02hVK5U2UPHULj88F0+7cDBrZ
uIt4ImfkabBoxTzkbFpG583H+u/E7Eu9aqSs/cwoUe+StCmrqzWaTOTECMYmzPhp
n+Sc8CnTXPnGFwIeI8MgwT0PPzhAsP6CRDlqWhqKa2NYOLQV07YRaXseVO6MGiKs
cpc/I1mbySKewQdPzH/iV8oScLumZfNpdWO9lfsbl83kqK/20U6o2YpxJM02PbyW
xPFsq47lw1uKA2wDrXKUXt4FMMgLG3/7FFXhEZn91QqhngLjYl/rNUssuHLoPj1P
rCy7Lobio3aP5ZMqz6WryFyNSwb/EkaseMsUBzXgqd+L6a8VTxaJW732jcZZroiF
DsGJ6x9nxUWO/203Nit4ZoORUSS9/1F3dmKh7Gc+PoGD4FapUB8fepmrY7+EF3fx
DTvf95xhszWYijqy7DwaNz9+j5LP2RIUZNoQAhVB/0/E6xyjyfqZ90bp4RjZsbgy
LcsUDFDYg2WD7rlcz8sFWkz6GZdr1l0T08JcVLwyc6B49fFtHsufpaafItzRUZ6C
eWRgKRM+o/1Pcmqr4rTluCRVLERLiohEnMqE0yo7AgMBAAGjQjBAMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSCIS1mxteg4BXrzkWJ
d8RgnlRuAzANBgkqhkiG9w0BAQsFAAOCAGEAyN4auOfyYILVAzOBywaK8SJJ6ejq
kX/GM15oGQOGO0MBzwdw5AgeZYWR5hEit/UCI46uuR59H35s5r0l1ZUa8gWmr4UC
b6741jH/JclKyMeKqdmfS0mbEVeZkkMR3rYzpMzXjWR91M08KCy0mpbqTfXERMQL
qiCA2CIV9+BB/AYm/7k29UMUA2Z44RGx2iBfRGb4ACGIHgAoYXhVqAEBj500mv/0
OJD7uNGzcgBjCeaBxXntC6Z58hMLnPddDnskk7RI24Zf3lCGeOdA5jGokHZwYa+c
NywRtYK3qq4kNFtyDGkNzVmf9nGvnAvRCjj5BiKDUyUM/FHE5r7iOZULJK2v0ZXk

PwZpVnPDimZI+ymBV3QGypzqKOG4ZyYr8dW1P2WT+DZdjo2NQCCHGervJ8A9tDkP
JXtoUHRVnAxZfVo9QZQlUgigRywVMRnVvwdVxrsStZf0X4OFunHB2WyBEXYKCrC/
gpf36j36+uwtqSiUO1bd0lEursC9CBWMD1I0ltabrNMdjmEPNXubrlpC2JgQCA2
j6/7Nu4tCEoduL+bXPjqpRugc6bY+G7gMwRfaKonh+3ZwZCc7b3jajWvY9+rGNm6
5ulK6lCKD2GTHultGelwldWSXQ62B68ZgI9HkFFLLk3dheLSCIIF5r8GrBQAuUB
o2M3IUxExJtRmREOc5wGj1QupyheRdMHVi03vYVElOEMSsyycw5KFNGHLD7ibSkNS
/jQ6fbjpKdx2qcgw+BRxgMYeNkh0IkFch4LoGHGLQYIE535YW6i4jRPpp2zDR+2z
Gp1iro2C6pSe3VkQw63d4k3jMdXH7OjysP6SHhYKGvzZ8/gntsm+HbRsZJB/9OTE
W9c3rkIO3aQab3yIVMUWbuF6aC74Or8NpDyJO3inTmODBCeIZ43ygknQW/2xzQ+D
hNQ+IIX3Sj0rnP0qCglN6oH4EZw=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICkzCCAbGgAwIBAgIKe3G2gla4EnycqDAKBggqhkJOPQQDAzBaMQswCQYDVQQG
EwJVUzETMBEGA1UECXMkZW1TaWduIFBLSTEUMBIGA1UEChMLZU11ZGhyYSBJbmMx
IDAeBgNVBAMTF2VtU2lnbiBFQ0MgUm9vdCBDQSAiIEMzMB4XDTE4MDIxODE4MzAw
MFoXDTE4MDIxODE4MzAwMFowWjELMAkGA1UEBhMCVVMxEzARBgNVBASTCmVtU2ln
biBQS0kxZDASBgNVBAoTC2VNdWRocmEgSW5jMSAwHgYDVQQDEXdibVNPZ24gRUND
IFJvb3QgQ0EgLSBDMzB2MBAGByqGSM49AgEGBSuBBAAiA2IABP2lYa57JhAd6bci
MK4G9IGzsUJxlTm801Ljr6/58pc1kjZGDoeVjbk5Wum739D+yAdBPLtVb4Ojavti
sIGJAnB9SMVK4+kiVCJNk7tCDK93nComfddhEc5lx/h//vXyqaNCMEAwHQYDVR0O
BBYEFptaSNCAIEDyqOkAB2kZd6fmw/TPMA4GA1UdDwEB/wQEAWIBBjAPBgNVHRMB
Af8EBTADAQH/MAoGCCqGSM49BAMDA2gAMGUCMQC02C8Cif22TGK6Q04ThHK1rt0c
3ta13FaPWEBaLd4gTCKDypOofu4SQMfWh0/434UCMBwUZOR8loMRnLDRWmFLpg9J
0wD8ofzkpf9/rdcw0Md3f76BB1UwUCAU9Vc4CqgxUQ==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICtjCCAdOgAwIBAgIKPPYHqWhwDtqLhDAKBggqhkJOPQQDAzBrMQswCQYDVQQG
EwJJTjETMBEGA1UECXMkZW1TaWduIFBLSTEIMCMGA1UEChMcZU11ZGhyYSBUZWNo
bm9sb2dpZXMGtGltXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0g
RzMwHhcnMTgwMjE4MTgzMDAwWhcNNDMwMjE4MTgzMDAwWjBrMQswCQYDVQQGEwJJ
TjETMBEGA1UECXMkZW1TaWduIFBLSTEIMCMGA1UEChMcZU11ZGhyYSBUZWNoZm9s
b2dpZXMGtGltXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0gRzMw
djAQBgcqhkJOPQIBBgUrgQQAIGNiAAQjPQy4LRL1KPOxst3iAhKAnjlfSU2fySU0
WXTsuwYc58Byr+iuL+FBVlcUqEqy6HyC5ltqtdyzdc6LBtCGI79G1Y4PPwT01xyS
fvalY8L1X44uT6EYgQIrMgqCZH0Wk9GjQjBAMB0GA1UdDgQWBBR8XQKEE9TMipuB
zhccLikenEhjQjAObgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQH/BAUwAwEB/zAKBggq
hkjOPQQDAwNpADBmAjEAvvNhzwIQHWSVB7gYboiFBS+DCBeQyh+KTOgNG3qxrWB
CUfvO6wIBHxcmbHtRwfSAjEanbpV/KIK603t5nYBQnvI+GDZjVGLVTv7jHvrZQnD
+JbNR6iC8hZVdyR+EhCVBCyj

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDczCCAlugAwIBAgILAK7PALrEzzL4Q7IwDQYJKoZIhvcNAQELBQAwVjELMAkG

A1UEBhMCVVMxEzARBgNVBA5TCmVtU2lnbiBQs0kxFDASBgNVBAoTC2VNdWRocmEg
SW5jMRwwGgYDVQQDExNlbVNpZ24gUm9vdCBDQSAiIEMxMB4XDTE4MDIxODE4MzAw
MFoXDTQzMdIxODE4MzAwMFowVjELMAkGA1UEBhMCVVMxEzARBgNVBA5TCmVtU2ln
biBQs0kxFDASBgNVBAoTC2VNdWRocmEgSW5jMRwwGgYDVQQDExNlbVNpZ24gUm9v
dCBDQSAiIEMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz+upufGZ
BczYKCFK83M0UYRWEPWgTywS4/oTmifQz/l5GnRfHXk5/Fv4cI7gkL35CX5VIPZ
HdPIWoU/Xse2B+4+wM6ar6xWQio5JXDWv7V7Nq2s9nPczdcdioOI+yuQFTdrHCZH
3DspVpNqs8FqOp099cGXOFgFixwR4+S0uF2FHYP+eF8LRWgYSKVGczQ7/g/IdrvH
GPMF0Ybzhe3nudkyrVWlZqa2kbBPrH4VI5b2P/AgNBbeCsbEBEV5f6f9vtKppa+c
xSMq9zwhbL2vj07FOrLzNBL834AaSaTUqZX3noleoomslMuoaJuvimUnzYnu3Yy1
aylwQ6BpC+S5DwIDAQABo0IwQDAdBgNVHQ4EFgQU/qHgCB4qAzlSWkK+XJGFehiq
TbUwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBAMJKVvoVIXsoounlHfv4LcQ5lkFMOycsxGwYFYDGrK9HWS8mC+M2sO87
/kOXSTKZEhVb3xEx/6tT+LvBeA+snFOvV71ojD1pM/CjoCNjO2RnIkSt1XHLVip4
kqNPEjE2NuLe/gDEo2APJ62gsIq1NnpSob0n9CAnYuhNlCQT5AoE6TyrLshDCURG
YQTISTR+08TI9Q/Aqum6VF7zYytPT1DU/r17mYw9wC68AivTxEDkigcxHpvOJpkT
+xHqmiIMERNHXhuBUDDIlhJu58tBf5E7oke3VIAb3ADMmpDqw8NQBmIMMMAVSKeo
WXzhriKi4gp6D/piq1JM4fHfyr6DDUI=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDIDCCAnygAwIBAgIKMfXkYgxsWO3W2DANBgkqhkiG9w0BAQsFADBnMQswCQYD
VQQGEwJTTjETMBEGA1UECXMkZW1TaWduIFBLSSTEIMCMGA1UEChMcZU11ZGhyYSBU
ZWNobm9sb2dpZXMGtGltXRlZDEcMBoGA1UEAxMTZW1TaWduIFJvb3QgQ0EgLSBH
MTAeFw0xODAyMTgxODMwMDBaFw00MzAyMTgxODMwMDBaMGcxGzAJBgNVBAYTAklO
MRMwEQYDVQQLZWplbVNpZ24gUETJMSUwIwYDVQQKExxlTXVkaHJhIFRlY2hub2xv
Z2llcyBMAW1pdGVkMRwwGgYDVQQDExNlbVNpZ24gUm9vdCBDQSAiIEMxMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAk0u76WaK7p1b1TST0Bsew+eeuGQz
f2N4aLTNLnf115sgxk0pvLZoYIr3IZpWNVrzd3YzZr/k1ZLpVkJGoZM0Kd0WNHVO
8oG0x5ZOrRkVUkr+PHB1cM2vK6sVmJm8qrOLqs1D/fXqcP/tzxE7lM5OMhbTl0Aq
d7OvPAEsbO2ZLIvZTmmYsvePQbAyeGHWDV/D+qJakh1cF+ZwPjXnorfCYuKrpDhM
tTk1b+oDafo6VGiFbddyL0NVHpeNDtjVaqSW0RM8LHhQ6DqS0hdW5TUaQBw+jSzt
Od9C4INBdN+jzcKGYEho42kLVACL5HZpIQ15TjQIXhTCzLG3rdd8cIrHhQIDAQAB
o0IwQDAdBgNVHQ4EFgQU++8Nhp6w492pufEhF38+/PB3KxowDgYDVR0PAQH/BAQD
AgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAFn/8oz1h31x
PaOfG1vR2vjTnGs2vZupYeveFix0PZ7mdrXuqe8QhfnPZHr5X3dPpzzx5KsbEjM
wiI/aTvFthUvozXGaCocV685743QNcMYDHsAVhzNixl03r4PEuDQqqE/AjSxcM6d
GNYIAwlg7mDgfrbESQRRfXBgvKqy/3lyeqYdPV8q+Mri/Tm3R7nrft8EI6/6nAYH
6ftjk4BAtcZsCjEozgyfz7MjNYBBjWzEN3uBL4ChQEKF6dk4jeihU80Bv2noWgby
RQuQ+q7hv53yrlc8pa6yVvSLZUDp/TGBLPQ5Cdjua6e0ph0VpZj3AYHYhX3zUVxx
iN66zB+Afk0=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICDzCCAZWgAwIBAgIUbmq8WapTvpq5Z6LSa6Q75m0c1towCgYIKoZIzj0EAwMw
RzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28uLEx0ZC4xGjAY

1.40 busybox 1.37.0-r19

1.40.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2006 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must
not be misrepresented
as being the original software.
4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the

notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and
copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.
- c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution
of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.41 shadow 4.17.3-r0

1.41.1 Available under license :

SPDX-License-Identifier: BSD-3-Clause

All files under this project either

1. fall under the BSD 3 clause license (by default).
2. carry an SPDX header declaring what license applies.

or

3. list a full custom license

This software is originally

* Copyright (c) 1989 - 1994, Julianne Frances Haugh

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the copyright holders or contributors may not be used to

* endorse or promote products derived from this software without

* specific

prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A

* PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

* HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License

All new source code committed to the shadow project is assumed to be made available under the [BSD-3-Clause](../COPYING) license unless the submitter specifies another license at that time. The shadow maintainers reserve the right to refuse a submission if the license is deemed incompatible with the goals of the project.

****Note****: old code may be made available under another license, check the license tag for each file to get additional information.

1.42 libncursesw 6.5_p20250503-r0

1.42.1 Available under license :

Upstream source <https://invisible-island.net/ncurses/Ada95.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package

Copyright: 2010-2023,2024 by Thomas E. Dickey

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh

Copyright: 1994 X Consortium

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except

as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make` implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package

Copyright: 2003-2023,2024 by Thomas E. Dickey

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,

distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script `install-sh` is preferred over `install.sh`, to prevent
'make' implicit rules from creating a file called `install` from it
when there is no `Makefile`.

This script is compatible with the BSD `install` script, but was written
from scratch. It can only install one file at a time, a restriction
shared with many OS's `install` programs.

On Debian systems, the complete text of the GNU General
Public License can be found in `/usr/share/common-licenses/GPL-2'`

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses.html>

This package is used for testing builds of `ncurses`.

Current `ncurses` maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: `aclocal.m4` package

Copyright: 1996-2023,2024 by Thomas E. Dickey

License: X11

Files: `doc/html/NCURSES-Programming-HOWTO.html`

Copyright: 2001 by Pradeep Padala

License: X11

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, distribute with modifications, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software
is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files:

install-sh

Copyright: 1994 X Consortium

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Files: progs/tset.c ncurses/tinfo/read_termcap.c

Copyright: 1980,1991,1992,1993 The Regents of the University of California.

License: BSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright

- notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-- vile: txtmode file-encoding=utf-8

Copyright 2018-2023,2024 Thomas E. Dickey

Copyright 1998-2017,2018 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the

sale, use or other dealings in this Software without prior written authorization.

```
-- vile:txtmode fc=72
-- $Id: COPYING,v 1.13 2024/01/05 21:13:17 tom Exp $
```

1.43 ncurses 6.5_p20250503-r0

1.43.1 Available under license :

```
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

buildddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
```

```

_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/${_pkgreal}-${pkgver}"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"

```



```

source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="${srcdir}/${_pkgname}-${pkgver}

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=${pkgver}-r${pkgrel}" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

#ifdef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else

```

```

# define __THROW
# define __NTH(fct)  fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

```

- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- *
- * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
- * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
- * IN NO EVENT SHALL THE
- AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
- * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
- * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
- * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
- * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- */

```
#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_
```

```
#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
 (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif
```

```
#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif
```

```
/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
data structure. Every operation
* on the tree causes a splay to happen. The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree. On the other hand,
* every lookup causes memory writes.
```

- *
 - * The Balance Theorem bounds the total access time for m operations
 - * and n inserts on an initially empty tree as $O((m + n)\lg n)$. The
 - * amortized cost for a sequence of m accesses to a splay tree is $O(\lg n)$;
- *
 - * A red-black tree is a binary search tree with the node color as an
 - * extra attribute. It fulfills a set of conditions:
 - * - every search path from the root to a leaf consists of the
 - * same number of black nodes,
 - * - each red node (except for the root) has a black parent,
 - * - each leaf node is black.
- *
 - * Every operation on a red-black tree is bounded as $O(\lg n)$.
 - * The maximum height of a red-black tree is $2\lg(n+1)$.
- */

```
#define SPLAY_HEAD(name, type)  \
struct name {                  \
    struct type *sph_root; /* root of the tree */ \
}
```

```
#define SPLAY_INITIALIZER(root) \
{ NULL }
```

```
#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}
```

```
#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)
```

```
/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root,
```

```

field) = SPLAY_LEFT(tmp, field); \
SPLAY_LEFT(tmp, field) = (head)->sph_root; \
(head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \

```

```

name##_SPLAY(head, elm); \
if (SPLAY_RIGHT(elm, field) != NULL) { \
    elm = SPLAY_RIGHT(elm, field); \
    while (SPLAY_LEFT(elm, field) != NULL) { \
        elm = SPLAY_LEFT(elm, field); \
    } \
} \
else \
    elm = NULL; \
return (elm); \
} \
\
static __sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\

```

```

struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
        return (elm); \
    } \
    return (NULL); \
} \
\
void \
name##_SPLAY(struct \
    name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \

```

```

    if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
        break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp > 0) { \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

```

```

#define SPLAY_NEGINF -1

```



```

#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
    struct name { \
        struct type *rbh_root; /* root of the tree */ \
    }

#define RB_INITIALIZER(root) \
    { \
        NULL \
    }

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
    struct { \
        struct type *rbe_left; /* left element */ \
        struct type *rbe_right; /* right element */ \
        struct type *rbe_parent; /* parent element */ \
        int rbe_color; /* node color */ \
    }

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \

```

```

RB_PARENT(elm, field) = parent; \
RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
    != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \

```

```

(head)->rbh_root = (tmp); \
RB_RIGHT(tmp, field) = (elm); \
RB_PARENT(elm, field) = (tmp); \
RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
        type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.
 * Moves node close to the key of elm to top
 */
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
    name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
    { \
        struct type \
        *parent, *gparent, *tmp; \
        while ((parent = RB_PARENT(elm, field)) != NULL && \
            RB_COLOR(parent, field) == RB_RED) { \
            gparent = RB_PARENT(parent, field); \
            if (parent == RB_LEFT(gparent, field)) { \
                tmp = RB_RIGHT(gparent, field); \
                if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                    RB_COLOR(tmp, field) = RB_BLACK; \
                    RB_SET_BLACKRED(parent, gparent, field); \
                    elm = gparent; \

```

```

    continue; \
} \
if (RB_RIGHT(parent, field) == elm) { \
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
    tmp = RB_LEFT(gparent, field); \
    if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
        RB_COLOR(tmp, field) = RB_BLACK; \
        RB_SET_BLACKRED(parent, gparent, field);\
        elm = gparent; \
        continue; \
    } \
    if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
    RB_SET_BLACKRED(parent, gparent, field); \
    RB_ROTATE_LEFT(head, gparent, tmp, field); \
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \

```

```

    RB_COLOR(RB_RIGHT(tmp,
field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
    if (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
        struct type *oleft; \
        if ((oleft = RB_LEFT(tmp, field)) \
            != NULL) \
            RB_COLOR(oleft, field) = RB_BLACK;\
        RB_COLOR(tmp, field) = RB_RED; \
        RB_ROTATE_RIGHT(head, tmp, oleft, field);\
        tmp = RB_RIGHT(parent, field); \
    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_RIGHT(tmp, field)) \
        RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field); \
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\

```

```

    tmp = RB_LEFT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_LEFT(tmp, field)) \
    RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_RIGHT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
= child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
}

```

```

if (RB_PARENT(old, field)) { \
    if (RB_LEFT(RB_PARENT(old, field), field) == old)\
        RB_LEFT(RB_PARENT(old, field), field) = elm;\
    else \
        RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else \
    RB_ROOT(head) = elm; \
RB_PARENT(RB_LEFT(old, field), field) = elm; \
if (RB_RIGHT(old, field)) \
    RB_PARENT(RB_RIGHT(old, field), field) = elm; \
if (parent) { \
    left = parent; \
    do { \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
} \
goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent, field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \

```

```

    comp = (cmp)(elm, parent); \
    if (comp < 0) \
        tmp = RB_LEFT(tmp, field); \
    else if (comp > 0) \
        tmp = RB_RIGHT(tmp, field); \
    else \
        return (tmp); \
} \
RB_SET(elm, parent, field); \
if (parent != NULL) { \
    if (comp < 0) \
        RB_LEFT(parent, field) = elm; \
    else \
        RB_RIGHT(parent, field) = elm; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head, \
elm); \
return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \

```



```

    comp = cmp(elm, tmp); \
    if (comp < 0) { \
        res = tmp; \
        tmp = RB_LEFT(tmp, field); \
    } \
    else if (comp > 0) \
        tmp =
RB_RIGHT(tmp, field); \
    else \
        return (tmp); \
    } \
    return (res); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_NEXT(struct type *elm) \
{ \
    if (RB_RIGHT(elm, field)) { \
        elm = RB_RIGHT(elm, field); \
        while (RB_LEFT(elm, field)) \
            elm = RB_LEFT(elm, field); \
    } else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
            elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \
        elm = RB_LEFT(elm, field); \
        while (RB_RIGHT(elm, field)) \
            elm = RB_RIGHT(elm, field); \
    } \
    else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \

```

```

    elm = RB_PARENT(elm, field); \
else { \
    while (RB_PARENT(elm, field) && \
        (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
        elm = RB_PARENT(elm, field); \
    elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x, y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
for ((x) = RB_MIN(name, head); \
    (x) != NULL; \
    (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
    ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
    (x) = (y))

```

```

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
    ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
    (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
    (x) != NULL; \
    (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
    (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name, \
head, y) \
for ((x) = RB_MAX(name, head); \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
    (x) = (y))

#endif /* _SYS_TREE_H_ */
MIT
# MIT License

```

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

arch="all"

options="!check" # No test suite.

license="LGPL-2.0-or-later"

subpackages="\$pkgname-dev"

source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-\$pkgver.tar.bz2"

"

builddir="\$srcdir"/libart-lgpl-\$pkgver

```
prepare() {  
    update_config_sub  
    default_prepare  
}
```

```
build() {  
    ./configure \  
        --build=$CBUILD \  
        --host=$CHOST \  
        --prefix=/usr  
    make  
}
```

```
package() {  
  make DESTDIR="$pkgdir" install  
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7  
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"  
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.44 utmps-libs 0.1.3.1-r0

1.44.1 Available under license :

Copyright (c) 2018-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN

ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.45 iputils-ping 20240905-r0

1.45.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such

interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution

and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing
or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing

version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.46 libmagic 5.46-r2

1.46.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you

conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.47 libcrypto3 3.5.4-r0

1.47.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy,

distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use
- in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.
- d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the

Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of

the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate

manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.48 alpine-baselayout-data 3.7.0-r0

1.48.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=""
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/_pkgreal-$pkgver.tar.gz"

_buildddir="$srcdir/_pkgreal-$pkgver"

prepare() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test
}

package() {
    cd "$_buildddir"
    make
    DESTDIR="$pkgdir" install || return 1
    find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
```

```
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"
```

```
_builddir="$srcdir"/libart_lgpl-$pkgver
```

```
prepare() {
    cd "$_builddir"
    update_config_sub || return 1
}
```

```
build() {
    cd "$_builddir"
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        || return 1
    make || return 1
}
```

```
package() {
    cd "$_builddir"
    make DESTDIR="$pkgdir" install || return 1
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
```

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition, Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process,

software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.

3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect

is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.

/*

* Copyright (c) 1983, 1988 Regents of the University of California.

* All rights reserved.

*

* Redistribution and use in source and binary forms are permitted

* provided that the above copyright notice and this paragraph are

* duplicated in all such forms and that any documentation,

* advertising materials, and other materials related to such

* distribution and use acknowledge that the software was developed

* by the University of California, Berkeley. The name of the
* University may not be used to endorse or promote products derived
* from this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

*/

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Valery Kartel <valery.kartel@gmail.com>
pkgname=perl-bsd-resource
_pkgreal=BSD-Resource
pkgver=1.2909
pkgrel=2
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="http://search.cpan.org/dist/BSD-Resource/"
arch="all"
license="GPL PerlArtistic"
cpandepends=""
cpanmakedepends=""
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/${_pkgreal}-${pkgver}.tar.gz"
```

```
_builddir="$srcdir/${_pkgreal}-${pkgver}"
```

```
prepare() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}
```

```
build() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test || return 1
}
```

```
package() {
    cd "$_builddir"
    make DESTDIR="$pkgdir" install || return 1
    find "$pkgdir"
    \(-name perllocal.pod -o -name .packlist\) -delete
}
```

```
md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
```

```
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-
```

1.2909.tar.gz"
sha512sums="5f489eac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd
85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.

Your legal department tells

you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:

The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied

by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>

Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation, details at:

<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.

Contributor: Fabian Affolter <fabian@affolter-engineering.ch>

Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

```

pkgname=py-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.0
pkgrel=1
pkgdesc="Extension for flake8 which checks for copyrights"
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="flake8"
makedepends="python3-dev"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/${_pkgname}-${pkgver}

build() {
    cd "$builddir"
    python3 setup.py build || return 1
}

package() {
    cd "$builddir"
    python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1
}

md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-0.2.0.tar.gz"
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528fa5d586c3b1be9570f4a977a7e2417b586e543bfba12abce256b2023c8c4993eeb90flake8-copyright-0.2.0.tar.gz"
mini_sendmail - accept email on behalf of real sendmail

```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.49 github.com/hillu/go-yara/v4 4.3.4

1.49.1 Available under license :

Copyright (c) 2015-2020 Hilko Bengen <bengen@hilluzination.de>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.50 [iputils-tracepath](https://github.com/0x0905/iputils-tracepath) 20240905-r0

1.50.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in

themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals

of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than ``show w'` and ``show c'`; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.51 iputils 20240905-r0

1.51.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below,

refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but

does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source

code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software

Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General Public License instead of this License.

1.52 alpine-baselayout 3.7.0-r0

1.52.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=" "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/${_pkgreal}-${pkgver}.tar.gz"

_buildddir="$srcdir/${_pkgreal}-${pkgver}"

prepare() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test
}

package() {
    cd "$_buildddir"
    make
    DESTDIR="$pkgdir" install || return 1
    find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"
```

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"
```

```
_builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {
  cd "$_builddir"
  update_config_sub || return 1
}
```

```
build() {
  cd "$_builddir"
  ./configure \
    --build=$CBUILD \
    --host=$CHOST \
    --prefix=/usr \
    || return 1
  make || return 1
}
```

```
package() {
  cd "$_builddir"
  make DESTDIR="$pkgdir" install || return 1
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart-lgpl-2.3.21.tar.bz2"
```

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with

distribution. In addition,

Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United

States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights,

trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.

/*

* Copyright (c) 1983, 1988 Regents of the University of California.

- * All rights reserved.
- *
- * Redistribution and use in source and binary forms are permitted
- * provided that the above copyright notice and this paragraph are
- * duplicated in all such forms and that any documentation,
- * advertising materials, and other materials related to such
- * distribution and use acknowledge that the software was developed
- * by the University of California, Berkeley. The name of the
- * University may not be used to endorse or promote products derived
- * from this software without specific prior written permission.
- * THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
- * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

*/

```
# Automatically generated by apkgbuild-cpan, template 1
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Valery Kartel <valery.kartel@gmail.com>
pkgname=perl-bsd-resource
_pkgreal=BSD-Resource
pkgver=1.2909
pkgrel=2
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="http://search.cpan.org/dist/BSD-Resource/"
arch="all"
license="GPL PerlArtistic"
cpandepends=""
cpanmakedepends=" "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/${_pkgreal}-${pkgver}.tar.gz"
```

```
_builddir="$srcdir/${_pkgreal}-${pkgver}"
```

```
prepare() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}
```

```
build() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
    make && make test || return 1
}
```

```
package() {
    cd "$_builddir"
```

```

make DESTDIR="$pkgdir" install || return 1
find "$pkgdir"
\ ( -name perllocal.pod -o -name .packlist \) -delete
}

```

```

md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-
1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd
85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.

Your legal department tells

you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for

possible inclusion in the SQLite source tree, please accompany the patch with the following statement:
The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>
Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation, details at:
<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.

```
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.0
pkgrel=1
pkgdesc="Extension for flake8 which checks for copyrights"
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="flake8"
makedepends="python3-dev"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver
```

```
build() {
    cd "$builddir"
    python3 setup.py build || return 1
}
```

```
package() {
    cd "$builddir"
    python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1
}
```

```
md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-0.2.0.tar.gz"
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e2417b586e543bfba12abce256b2023c8c4993eeb90
flake8-copyright-0.2.0.tar.gz"
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.53 alpine-release 3.22.2-r0

1.53.1 Available under license :

MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

1.54 libbsd 0.12.2-r0

1.54.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Files:

include/bsd/err.h

include/bsd/stdlib.h

include/bsd/sys/param.h

include/bsd/unistd.h

src/bsd_getopt.c

src/err.c

src/fgetln.c

src/progname.c

Copyright:

Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>

Copyright 2005 Hector Garcia Alvarez

Copyright 2005 Aurelien Jarno

Copyright 2006 Robert Millan

Copyright 2018 Facebook, Inc.

License: BSD-3-clause

Files:

include/bsd/netinet/ip_icmp.h

include/bsd/sys/bitstring.h

include/bsd/sys/queue.h

include/bsd/sys/time.h

include/bsd/timeconv.h

include/bsd/vis.h

man/bitstring.3bsd

man/errc.3bsd

man/explicit_bzero.3bsd

man/fgetln.3bsd

man/fgetwln.3bsd

man/fpurge.3bsd

man/funopen.3bsd

man/getbsize.3bsd

man/heapsort.3bsd

man/nlist.3bsd

man/pwcache.3bsd

man/queue.3bsd

man/radixsort.3bsd

man/reallocarray.3bsd

man/reallocf.3bsd

man/setmode.3bsd
man/strmode.3bsd
man/strnstr.3bsd
man/strtoi.3bsd
man/strtou.3bsd
man/unvis.3bsd
man/vis.3bsd
man/wcslcpy.3bsd
src/getbsize.c
src/heapsort.c
src/merge.c
src/nlist.c
src/pwcache.c
src/radixsort.c
src/setmode.c
src/strmode.c
src/strnstr.c
src/strtoi.c
src/strtou.c
src/unvis.c

Copyright:

Copyright 1980, 1982, 1986, 1989-1994

The Regents of the University of California. All rights reserved.

Copyright 1992 Keith Muller.

Copyright 2001 Mike Barcroft <mike@FreeBSD.org>

.

Some code is derived from software contributed to Berkeley by the American National Standards Committee X3, on Information Processing Systems.

.

Some code is derived from software contributed to Berkeley by Peter McIlroy.

.

Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

.

Some code is derived from software contributed to Berkeley by Dave Borman at Cray Research, Inc.

.

Some code is derived from software contributed to Berkeley by Paul Vixie.

.

Some code is derived from software contributed to Berkeley by Chris Torek.

.

Copyright UNIX System Laboratories, Inc.

All or some portions of this file are derived from material licensed

to the University of California by American Telephone and Telegraph Co. or Unix System Laboratories, Inc. and are reproduced herein with the permission of UNIX System Laboratories, Inc.

License: BSD-3-clause-Regents

Files:

src/vis.c

Copyright:

Copyright 1989, 1993

The Regents of the University of California. All rights reserved.

.

Copyright 1999, 2005 The NetBSD Foundation, Inc.

All rights reserved.

License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:

include/bsd/libutil.h

Copyright:

Copyright 1996 Peter Wemm <peter@FreeBSD.org>.

All rights reserved.

Copyright 2002 Networks Associates Technology, Inc.

All rights reserved.

License: BSD-3-clause-author

Files:

man/timeradd.3bsd

Copyright:

Copyright

2009 Jukka Ruohonen <jruohonen@iki.fi>

Copyright 1999 Kelly Yancey <kbyanc@posi.net>

All rights reserved.

License: BSD-3-clause-John-Birrell

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.

THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/setproctitle.3bsd

Copyright:

Copyright 1995 Peter Wemm <peter@FreeBSD.org>

All rights reserved.

License: BSD-5-clause-Peter-Wemm

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author Peter Wemm.
5. Modifications may be freely made to this file providing the above conditions are met.

Files:

include/bsd/stringlist.h

man/arc4random.3bsd

man/fmtcheck.3bsd

man/humanize_number.3bsd

man/stringlist.3bsd

man/timeval.3bsd

src/fmtcheck.c

src/humanize_number.c

src/stringlist.c

src/strtonum.c

Copyright:

Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014

The NetBSD Foundation, Inc.

Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>

All rights reserved.

.

Copyright 2014

The NetBSD Foundation, Inc.

All rights reserved.

.

Some code was derived from software contributed to The NetBSD Foundation by Taylor R. Campbell.

.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.

.

Some code is derived from software contributed to The NetBSD Foundation by Jason R. Thorpe of the Numerical Aerospace Simulation Facility, NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.

.

Some code is derived from software contributed to The NetBSD Foundation by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

.

Some code is derived from software contributed to The NetBSD Foundation by Jukka Ruohonen.

License: BSD-2-clause-NetBSD

Files:

include/bsd/sys/endian.h

man/byteorder.3bsd

man/closefrom.3bsd

man/expand_number.3bsd

man/flopen.3bsd

man/getpeereid.3bsd

man/pidfile.3bsd

src/expand_number.c

src/pidfile.c

src/reallocf.c

src/timeconv.c

Copyright:

Copyright 1998, M. Warner Losh <imp@freebsd.org>

All rights reserved.

.

Copyright 2001 Dima Dorfman.

All rights reserved.

.

Copyright 2001 FreeBSD Inc.

All rights reserved.

.

Copyright 2002 Thomas Moestl <tmml@FreeBSD.org>

All rights reserved.

.

Copyright 2002 Mike Barcroft <mike@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Colin Percival

All rights reserved.

.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>

Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.

.

Copyright 2009 Advanced Computing Technologies LLC

All rights reserved.

.

Copyright 2011 Guillem Jover <guillem@hadrons.org>

License: BSD-2-clause

Files:

src/flopen.c

Copyright:

Copyright 2007-2009

Dag-Erling Codan Smrgrav

All rights reserved.

License: BSD-2-clause-verbatim

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/bsd/sys/tree.h

man/fparseln.3bsd

man/tree.3bsd

src/fparseln.c

Copyright:

Copyright 1997 Christos Zoulas.

All rights reserved.

.

Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

License: BSD-2-clause-author

Files:

include/bsd/readpassphrase.h

man/readpassphrase.3bsd

man/strncpy.3bsd

man/strtonum.3bsd

src/arc4random.c

src/arc4random_linux.h

src/arc4random_uniform.c

src/arc4random_unix.h

src/arc4random_win.h

src/closefrom.c

src/freezero.c

src/getentropy_aix.c

src/getentropy_bsd.c

src/getentropy_hpux.c

src/getentropy_hurd.c

src/getentropy_linux.c

src/getentropy_osx.c

src/getentropy_solaris.c

src/getentropy_win.c

src/readpassphrase.c

src/reallocarray.c

src/reallocarray.c

src/strlcat.c

src/strncpy.c

test/explicit_bzero.c

test/strtonum.c

Copyright:

Copyright 2004 Ted Unangst and Todd Miller

All rights reserved.

.

Copyright 1996 David Mazieres <dm@uun.org>

Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015

Todd C. Miller <Todd.Miller@courtesan.com>

Copyright 2004 Ted Unangst

Copyright 2004 Otto Moerbeek <otto@drijf.net>

Copyright 2008 Damien Miller <djm@openbsd.org>

Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>

Copyright 2013 Markus Friedl <markus@openbsd.org>

Copyright 2014 Bob Beck <beck@obtuse.com>

Copyright 2014 Brent Cook <bcook@openbsd.org>

Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>

Copyright 2014 Theo de Raadt <deraadt@openbsd.org>

Copyright 2014 Google Inc.

Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>

License:

ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/inet_net_pton.c

Copyright:

Copyright 1996 by Internet Software Consortium.

License: ISC-Original

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice

and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL

DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/setproctitle.c

Copyright:

Copyright 2010 William Ahern

Copyright 2012 Guillem Jover <guillem@hadrons.org>

License: Expat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

src/explicit_bzero.c

src/chacha_private.h

Copyright:

None

License: public-domain

Public domain.

Files:

man/mdX.3bsd

Copyright:

None

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this

file. As long as you retain this notice you

can do whatever you want with this stuff. If we meet some day, and you think

this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause-author

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGE.

1.55 confluentinc-confluent-kafka-go 2.12.0

1.55.1 Available under license :

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is

furnished

to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please

do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> /oo/

<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

LICENSE.lz4

src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

//-----

// MurmurHash2 was written by Austin Appleby, and is placed in the public

// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For

files in src/nanopb :
<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may

not be removed or altered from any source distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution

incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE

text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.queue

For

sys/queue.h:

- * Copyright (c) 1991, 1993
- * The Regents of the University of California. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 4. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR
- IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * @(#)queue.h 8.5 (Berkeley) 8/20/94
- * \$FreeBSD\$

LICENSE.regex

regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

```
-----  
#####  
# LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #  
# originally retrieved from http://github.com/andikleen/snappy-c #  
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #  
#####
```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinycthread

From <https://github.com/tinycthread/tinycthread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

/*

* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

*

* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
* ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN

* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

*

* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.

*/

/*_

* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.

*

* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice,
this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.

*

* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.

*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations,
or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems

that are managed by, or on behalf of, the

Licensors for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted

against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.56 ssl-client 1.37.0-r19

1.56.1 Available under license :

No license file was found, but licenses were detected in source scan.

+ * Licensed under GPLv2 or later, see file LICENSE in this source tree.

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0024-umount-Implement-O-option-to-unmount-by-mount-option.patch

No license file was found, but licenses were detected in source scan.

/*

* depmod - generate modules.dep

* Copyright (c) 2008 Bernhard Reutner-Fischer

- * Copyright (c) 2008 Timo Teras <timo.teras@iki.fi>

+ * Copyright (c) 2008-2015 Timo Teras <timo.teras@iki.fi>

* Copyright (c) 2008 Vladimir Dronnikov

*

* Licensed under GPLv2 or later, see file LICENSE in this source tree.

@@ -25,6 +25,24 @@

#include "modutils.h"

#include <sys/utsname.h> /* uname() */

+#define INDEX_MINCHAR 32

+#define INDEX_MAXCHAR 128

+

+typedef struct index_node {

+ char *prefix;

+ llist_t *values;

+ struct index_node *children[INDEX_MAXCHAR-INDEX_MINCHAR];

+} index_node;

+

+struct globals {

+ module_db db;

+ index_node *root_node;

+} FIX_ALIASING;

+#define G (*ptr_to_globals)


```

+ #define INIT_G() do { \
+ SET_PTR_TO_GLOBALS(xzalloc(sizeof(G))); \
+ } while (0)
+
+
+ /*
+ * Theory of operation:
+ * - iterate over all modules and record their full path
@@ -53,18 +71,12 @@ static int FAST_FUNC parse_module(struct recursive_state
*state,

for (ptr = image; ptr < image + len - 10; ptr++) {
if (is_prefixed_with(ptr, "depends=")) {
- char *u;
-
ptr += 8;
- for (u = ptr; *u; u++)
- if (*u == '-')
- *u = '_';
- ptr += string_to_llist(ptr, &e->deps, ",");
+ string_to_llist(replace_underscores(ptr), &e->deps, ",");
} else if (ENABLE_FEATURE_MODUTILS_ALIAS
&& is_prefixed_with(ptr, "alias=")
) {
- llist_add_to(&e->aliases, xstrdup(ptr + 6));
- ptr += strlen(ptr);
+ llist_add_to(&e->aliases, replace_underscores(xstrdup(ptr + 6)));
} else if (ENABLE_FEATURE_MODUTILS_SYMBOLS
&& is_prefixed_with(ptr, "__ksymtab_")
) {
@@ -74,9 +86,10 @@ static int FAST_FUNC parse_module(struct recursive_state *state,
) {
continue;
}
- llist_add_to(&e->symbols, xstrdup(ptr));
- ptr += strlen(ptr);
- }
+ llist_add_to(&e->symbols, xasprintf("symbol:%s", ptr));
+ } else
+ continue;
+ ptr += strlen(ptr);
}
free(image);

@@ -108,12 +121,6 @@ static void order_dep_list(module_db *modules, module_entry *start, llist_t
*add
}
}

```

```

-static void xfreopen_write(const char *file, FILE *f)
-{
- if (freopen(file, "w", f) == NULL)
- bb_perror_msg_and_die("can't open '%s'", file);
-}
-
//usage:#if !ENABLE_MODPROBE_SMALL
//usage:#define depmod_trivial_usage "[ -n ] [ -b BASE ] [ VERSION ] [ MODFILES ]..."
//usage:#define depmod_full_usage "\n\n"
@@ -167,6 +174,169 @@ enum {
OPT_C = (1 << 9), /* -C,--config etc_modules_conf: ignored */

```

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0018-depmod-support-generating-kmod-binary-index-files.patch

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2008 Natanael Copa <natanael.copa@gmail.com>
* All rights reserved.
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU General Public License version 2 as published
* by the Free Software Foundation. See http://www.gnu.org/ for details.
*
*/

```

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/bbsuid.c

1.57 libapk2 2.14.9-r3

1.57.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share

and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

(Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means

all

the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the

only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse

of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS

FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a
copy of the GNU General Public License along with
this program; if not, write to the Free Software Foundation, Inc., 51 Franklin
Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when
it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software,
and you are welcome to redistribute it under certain conditions; type `show
c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may be
called something other than `show w' and `show c'; they could even be mouse-clicks
or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school,
if any, to sign a "copyright
disclaimer" for the program, if necessary. Here
is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision'
(which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General
Public License does not permit incorporating your program into proprietary
programs. If your program is a subroutine library, you may consider it more
useful to permit linking proprietary applications with the library. If this
is what you want to do, use the GNU Lesser General Public License instead
of this License.

1.58 clientmodel 0.6.2

1.58.1 Available under license :

Data model artifacts for Prometheus.
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.59 skalibs-libs 2.14.4.0-r0

1.59.1 Available under license :

Copyright (c) 2011-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.60 musl-utils 1.2.5-r10

1.60.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Authors/contributors include:

A. Wilcox
Ada Worcester
Alex Dowad
Alex Suykov
Alexander Monakov
Andre McCurdy
Andrew Kelley
Anthony G. Basile
Aric Belsito
Arvid Picciani
Bartosz Brachaczek
Benjamin Peterson
Bobby Bingham
Boris Brezillon
Brent Cook
Chris Spiegel
Clment Vasseur
Daniel Micay
Daniel Sabogal
Daurnimator
David Carlier
David Edelsohn
Denys Vlasenko
Dmitry Ivanov
Dmitry V. Levin
Drew DeVault
Emil Renner Berthing
Fangrui Song
Felix Fietkau
Felix Janda

Gianluca Anzolin
Hauke Mehrrens
He X
Hiltjo Posthuma
Isaac Dunham
Jaydeep Patil
Jens Gustedt
Jeremy Huntwork
Jo-Philipp Wich
Joakim Sindholt
John Spencer
Julien Ramseier
Justin Cormack
Kaarle Ritvanen
Khem Raj
Kylie McClain
Leah Neukirchen
Luca Barbato
Luka
Perkov
M Farkas-Dyck (Strake)
Mahesh Bodapati
Markus Wichmann
Masanori Ogino
Michael Clark
Michael Forney
Mikhail Kremnyov
Natanael Copa
Nicholas J. Kain
orc
Pascal Cuoq
Patrick Oppenlander
Petr Hosek
Petr Skocik
Pierre Carrier
Reini Urban
Rich Felker
Richard Pennington
Ryan Fairfax
Samuel Holland
Segev Finer
Shiz
sin
Solar Designer
Stefan Kristiansson
Stefan O'Rear
Szabolcs Nagy
Timo Ters

Trutz Behn
Valentin Ochs
Will Dietz
William Haddon
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (`src/string/aarch64/*`) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (`src/crypt/crypt_des.c`) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/crypt/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsort.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the ellcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

1.61 github.com/aws/aws-sdk-go-v2/internal/ini 1.8.4

1.61.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody

```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

```

```

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization

```

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd

```

Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization
  OperationDeserializer
  InterceptBeforeDeserialization
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
  InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licenssemanager
```

```
import (
```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

```

```

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {

```

```

    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}

```

```

}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if

```



```

err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {

```

```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an

```

```

array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based

```

```

subscription.</p>",
    "smithy.api#required": {}
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
},

"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)

```

```

license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        },
    },

```



```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
      }
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                },
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                    ],
                },
            ],
            "assign": "true"
        }
    ],
    "rules": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
        }
    ],
}

```



```

        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```



```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```



```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```



```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
        "smithy.api#required": {}
    },
    "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```

"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }

```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}

```



```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

```

```

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${ github.base_ref }
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                  path: new-ref
                  fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
                  ${ github.sha } > refDiffFiles.txt
            - name: Get Target Files
              run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

```

```

- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```



```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the checkout license.
    LicenseArn
    *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}

```

```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"

```

```

smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);

```

```

err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use  
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the  
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more  
// information, see [Upgrade a workspace to Grafana Enterprise].  
//
```

```
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
```

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))  
(*AssociateLicenseOutput, error) {  
    if params == nil {  
        params = &AssociateLicenseInput{}
```

```
    }  
  
    result, metadata, err := c.invokeOperation(ctx,  
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*AssociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type AssociateLicenseInput struct {
```

```
    // The type of license to associate with the workspace.
```

```
    //
```

```
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free  
    // trials.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseType types.LicenseType
```

```
    // The ID of the workspace to associate the license with.
```

```

//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,

```

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Deletes the specified license.

```

```

func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```



```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }  
}
```

```
result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```


InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
    },
    "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

        "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
        },
    },
    "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
        },
    },
    "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\t\tsuccessful subscription data request.</p>"
        },
    },
    "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
        },
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
        },
    },
    "traits": {
        "smithy.api#output": {}
    },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"

```

```

    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      }
    }
  }
}

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    },
    "type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

{
  "conditions": [
    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {

```

```

        "ref": "PartitionResult"
    },
    "supportsFIPS"
]
}
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],

```



```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"rules": [
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```



```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
        "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Notequal</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```



```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.62 sirupsen-logrus 1.9.3

1.62.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.63 cJSON 1.7.19-r0

1.63.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.63.2 Available under license :

Apache License 2.0

Contributors

=====

Original Author:

- [Dave Gamble](<https://github.com/DaveGamble>)

Current Maintainer:

- [Max Bruckner](<https://github.com/FSMaxB>)

- [Alan Wang](<https://github.com/Alanscut>)

Contributors:

* [Ajay Bhargav](<https://github.com/ajaybhargav>)

* [Alexander Vasiljev](<https://github.com/AlexanderVasiljev>)

* [Alper Akcan](<https://github.com/alperakcan>)

* [Andrew Tang](<https://github.com/singku>)

* [Andy](<https://github.com/mlh0101>)

* [Anton Sergeev](<https://github.com/anton-sergeev>)

* [Ben Buck Nason](<https://github.com/bnason-nf>)

* [Bernt Johan Damslara](<https://github.com/bjda>)

* [Bob Kocisko](<https://github.com/bobkocisko>)

- * [Christian Schulze](https://github.com/ChristianSch)
- * [Casperinous](https://github.com/Casperinous)
- * [ChenYuan](https://github.com/zjuchenyuan)
- * [Debora Grosse](https://github.com/DeboraG)
- * [dieyushi](https://github.com/dieyushi)
- * [Dngwn Hung ()](https://github.com/DongwenHuang)
- * [Dominik](https://github.com/DL6ER)
- * [Donough Liu](https://github.com/ldm0)

*

- [Erez Oxman](https://github.com/erez-o)
- * Eswar Yaganti
- * [Evan Todd](https://github.com/etodd)
- * [Fabrice Fontaine](https://github.com/ffontaine)
- * Ian Mobley
- * Irwan Djadjadi
- * [hopper-vul](https://github.com/hopper-vul)
- * [HuKeping](https://github.com/HuKeping)
- * [IvanVoid](https://github.com/mpi3pak)
- * [Jakub Wilk](https://github.com/jwilk)
- * [Jiri Zouhar](https://github.com/loigu)
- * [Jonathan Fether](https://github.com/jfether)
- * [Joshua Arulsamy](https://github.com/jarulsamy)
- * [Julian Ste](https://github.com/julian-st)
- * [Julin Vsquez](https://github.com/juvasquezg)
- * [Junbo Zheng](https://github.com/Junbo-Zheng)
- * [Kevin Branigan](https://github.com/kbranigan)
- * [Kevin Sapper](https://github.com/sappo)
- * [Kyle Chisholm](https://github.com/ChisholmKyle)
- * [Linus Wallgren](https://github.com/ecksun)
- * [Luo Jin](https://github.com/Up-wind)
- * [Max](https://github.com/maebex)
- * [MaxBrandtner](https://github.com/MaxBrandtner)
- * [Mateusz Szafoni](https://github.com/raiden00pl)

*

Mike Pontillo

- * [miaoerduo](https://github.com/miaoerduo)
- * [mohawk2](https://github.com/mohawk2)
- * [Mike Jerris](https://github.com/mjerris)
- * [Mike Robinson](https://github.com/mhrobinson)
- * [Moorthy](https://github.com/moorthy-bs)
- * [myd7349](https://github.com/myd7349)
- * [NancyLi1013](https://github.com/NancyLi1013)
- * [Orri](https://github.com/sbvoxel)
- * Paulo Antonio Alvarez
- * [Pawe Malowany](https://github.com/PawelMalowany)
- * [Pawel Winogrodzki](https://github.com/PawelWMS)
- * [prefetchnta](https://github.com/prefetchnta)
- * [Rafael Leal Dias](https://github.com/rafaeldias)

- * [Randy](https://github.com/randy408)
- * [raiden00pl](https://github.com/raiden00pl)
- * [Robin Mallinson](https://github.com/rmallins)
- * [Rod Vagg](https://github.com/rvagg)
- * [Roland Meertens](https://github.com/rmeertens)
- * [Romain Porte](https://github.com/MicroJoe)
- * [SANJEEV BA](https://github.com/basanjeev)
- * [Sang-Heon Jeon](https://github.com/Intuition)
- * [Sayan Bandyopadhyay](https://github.com/saynb)
- *
- [Simon Sobisch](https://github.com/GitMensch)
- * [Simon Ricaldone](https://github.com/simon-p-r)
- * [Stoian Ivanov](https://github.com/sdrsdr)
- * [SuperH-0630](https://github.com/SuperH-0630)
- * [Square789](https://github.com/Square789)
- * [Stephan Gatzka](https://github.com/gatzka)
- * [Tony Langhammer](https://github.com/BigBrainAFK)
- * [Vemake](https://github.com/vemakereporter)
- * [vwvw](https://github.com/vwvw)
- * [warmsocks](https://github.com/warmsocks)
- * [Wei Tan](https://github.com/tan-wei)
- * [Weston Schmidt](https://github.com/schmidtw)
- * [xiaomianhehe](https://github.com/xiaomianhehe)
- * [yangfl](https://github.com/yangfl)
- * [yuta-oxo](https://github.com/yuta-oxo)
- * [Zach Hinds](https://github.com/zhinds)
- * [Zhao Zhixu](https://github.com/zhaozhixu)
- * [10km](https://github.com/10km)

And probably more people on [SourceForge](https://sourceforge.net/p/cjson/bugs/search/?q=status%3Aclosed-rejected+or+status%3Aclosed-out-of-date+or+status%3Awont-fix+or+status%3Aclosed-fixed+or+status%3Aclosed&page=0)

Also

thanks to all the people who reported bugs and suggested new features.

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.64 prometheus-client-model 0.6.2

1.64.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.65 prometheus-client 1.23.2

1.65.1 Available under license :

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Prometheus instrumentation library for Go applications

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>

<https://github.com/beorn7/perks>

Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein

See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format

<http://github.com/golang/protobuf/>

Copyright 2010 The Go Authors

See source code for license details.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.66 xxhash 2.3.0

1.66.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.67 procs 0.19.2

1.67.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

1.68 stdlib go1.25.4

1.68.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style  
// license that can be found in the LICENSE file.
```

```
package copyright
```



```

import (
    "bytes"
    "internal/testenv"
    "io"
    "io/fs"
    "os"
    "path/filepath"
    "testing"
)

var copyright = []byte("Copyright")

var permitted = [][]byte{
    []byte("// Code generated by "),
    []byte("// Code generated from "),
    []byte("// Created by cgo -cdefs"),
    []byte("// DO NOT EDIT\n// generated by:"),
    []byte("// Empty assembly file"),
    []byte("// Generated using cgo"),
    []byte("// Original source:\n//\thttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
    []byte("// created by cgo -cdefs"),
    []byte("// go run mkasm.go"),
    []byte("// mkerrors"),
    []byte("// mksys"),
    []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}

func TestCopyright(t *testing.T) {
    buf := make([]byte, 2048)
    filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
        d fs.DirEntry, err error) error {
        if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
            return filepath.SkipDir
        }
        switch filepath.Ext(d.Name()) {
        default:
            return nil
        case ".s", ".go":
            // check
        }

        f, err := os.Open(path)
        if err != nil {
            t.Error(err)
            return nil
        }
        defer f.Close()
    }
}

```

```

n, err := f.Read(buf)
if err != nil && err != io.EOF {
    t.Error(err)
    return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
    return nil
}
for _, ok := range permitted {
    if bytes.HasPrefix(b, ok) {
        return nil
    }
}
t.Errorf("%s: missing copyright notice", path)
return nil
})
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at <https://boringssl.googlesource.com/boringssl/+/-/fips-20220613/LICENSE>.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: <https://cla.developers.google.com/clas>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

/* =====

* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following

disclaimer in

- * the documentation and/or other materials provided with the
- * distribution.
- *
- * 3. All advertising materials mentioning features or use of this
- * software must display the following acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- *
- * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
- * endorse or promote products derived from this software without
- * prior written permission. For written permission, please contact
- * openssl-core@openssl.org.
- *
- * 5. Products derived from this software may not be called "OpenSSL"
- * nor may "OpenSSL" appear in their names without prior written
- * permission of the OpenSSL Project.
- *
- * 6. Redistributions of any form whatsoever must retain the following
- * acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit
- * (<http://www.openssl.org/>)"
- *
- * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- * OF THE POSSIBILITY OF SUCH DAMAGE.
- * =====
- *
- * This product includes cryptographic software written by Eric Young
- * (ey@cryptsoft.com). This product
- * includes software written by Tim
- * Hudson (tjh@cryptsoft.com).
- *
- * /

Original SSLeay License

```

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of
 * the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are
 * not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

```

```

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
  POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

```

ISC license used for completely new code in BoringSSL:

```

/* Copyright (c) 2015, Google Inc.
*
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE
  OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

```

Some files from Intel carry the following license:

```

# Copyright (c) 2012, Intel Corporation
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
#
# * Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the
#   distribution.

```

```

#
# * Neither the name of the Intel Corporation nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
# ""AS IS"" AND ANY
# EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
# CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
# PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
# PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# Names should be added to this file as:
#   Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Andrea Leofreddi.

1.69 golang 1.25.4

1.69.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style  
// license that can be found in the LICENSE file.
```

```
package copyright
```

```
import (  
    "bytes"  
    "internal/testenv"  
    "io"  
    "io/fs"  
    "os"  
    "path/filepath"  
    "testing"  
)
```

```
var copyright = []byte("Copyright")
```

```

var permitted = [][]byte{
    []byte("// Code generated by "),
    []byte("// Code generated from "),
    []byte("// Created by cgo -cdefs"),
    []byte("// DO NOT EDIT\n// generated by:"),
    []byte("// Empty assembly file"),
    []byte("// Generated using cgo"),
    []byte("// Original source:\n/\thttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
    []byte("// created by cgo -cdefs"),
    []byte("// go run mkasm.go"),
    []byte("// mkerrors"),
    []byte("// mksys"),
    []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}

```

```

func TestCopyright(t *testing.T) {
    buf := make([]byte, 2048)
    filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
    d fs.DirEntry, err error) error {
        if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
            return filepath.SkipDir
        }
        switch filepath.Ext(d.Name()) {
        default:
            return nil
        case ".s", ".go":
            // check
        }
    }
}

```

```

f, err := os.Open(path)
if err != nil {
    t.Error(err)
    return nil
}
defer f.Close()
n, err := f.Read(buf)
if err != nil && err != io.EOF {
    t.Error(err)
    return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
    return nil
}
for _, ok := range permitted {
    if bytes.HasPrefix(b, ok) {
        return nil
    }
}

```

```

}
t.Errorf("%s: missing copyright notice", path)
return nil
})
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../..../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at <https://boringssl.googlesource.com/boringssl/+fips-20220613/LICENSE>.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:

<https://cla.developers.google.com/clas>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

```
/* =====
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following
disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
```

- * 5. Products derived from this software may not be called "OpenSSL"
- * nor may "OpenSSL" appear in their names without prior written
- * permission of the OpenSSL Project.
- *
- * 6. Redistributions of any form whatsoever must retain the following
- * acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit
- (<http://www.openssl.org/>)"
- *
- * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- * OF THE POSSIBILITY OF SUCH DAMAGE.
- * =====
- *
- * This product includes cryptographic software written by Eric Young
- * (eay@cryptsoft.com). This product
- includes software written by Tim
- * Hudson (tjh@cryptsoft.com).
- *
- */

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
```

- * Copyright remains Eric Young's, and as such any Copyright notices in
- * the code are not to be removed.
- * If this package is used in a product, Eric Young should be given attribution
- * as the author of
- the parts of the library used.
- * This can be in the form of a textual message at program startup or
- * in documentation (online or textual) provided with the package.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 3. All advertising materials mentioning features or use of this software
- * must display the following acknowledgement:
- * "This product includes cryptographic software written by
- * Eric Young (eay@cryptsoft.com)"
- * The word 'cryptographic' can be left out if the routines from the library
- * being used are
- not cryptographic related :-).
- * 4. If you include any Windows specific code (or a derivative thereof) from
- * the apps directory (application code) you must include an acknowledgement:
- * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
- *
- * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * The licence and distribution terms for any publically available version or
- * derivative of this code cannot be changed. i.e. this code cannot simply be
- * copied and put under another distribution licence
- * [including the GNU Public Licence.]
- */

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.

*

* Permission to use, copy, modify, and/or distribute this software for any

* purpose with or without fee is hereby granted, provided that the above

* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES

* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY

* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION

* OF CONTRACT, NEGLIGENCE

OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN

* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation

#

All rights reserved.

#

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

#

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

#

* Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the

distribution.

#

* Neither the name of the Intel Corporation nor the names of its

contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

#

#

THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION

""AS IS"" AND ANY

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.

<https://developers.google.com/open-source/cla/individual>
<https://developers.google.com/open-source/cla/corporate>

Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyouun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE

GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the
authors and should not be interpreted as representing official policies, either expressed
or implied, of Andrea Leofreddi.

1.70 [github.com/aws/aws-sdk-go-](https://github.com/aws/aws-sdk-go-v2/aws/protocol/eventstream) [v2/aws/protocol/eventstream](https://github.com/aws/aws-sdk-go-v2/aws/protocol/eventstream) 1.7.3

1.70.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks out the specified license for offline use.
```

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
```

```
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
    c.addOperationCheckoutBorrowLicenseMiddlewares)
```

```
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.

```

```

Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {

```

```

return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
}

```

```

    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = addRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options);
err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutBorrowLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```



```

"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {

```

```

    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}

```

```

    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "GetLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.

```

```

Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {

```

```

    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
}

```

```

    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = addRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack,
options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```


ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

```

```

type DeleteLicenseInput struct {

```

```

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

```

```

    noSmithyDocumentSerde
}

```

```

type DeleteLicenseOutput struct {

```

```

    // Date when the license is deleted.
    DeletionDate *string

```

```

    // License status.
    Status types.LicenseDeletionStatus

```

```

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}

```

```

if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,

```

```

    OperationName: "DeleteLicense",
  }
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",

```

```

"members": {
  "message": {
    "target": "smithy.api#String"
  }
},
"traits": {
  "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
  "smithy.api#error": "client"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user

```


administration in the Active Directory.</p>"

```
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}
```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
      },
    }
  }
}

```

```

    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        }
    ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {

```

```

"type": "structure",
"members": {
  "LicenseServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {

```



```

    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider

```

```

operation.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#output": {}
},
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
  ],

```

```

    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},

"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
  "type": "structure",
  "members":
{
  "Attribute": {
    "target": "smithy.api#String",
    "traits": {

```

```

    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
  },
  "Operation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
    }
  },
  "Value": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Value of the filter.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
    }
},
"Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

        "smithy.api#required": {}
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
    }
},
"Status": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
    }
},
"IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
},
"FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
},
"OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    },
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  }

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
      },
    ]
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {

```

```

"version": "1.0",
"parameters": {

  "Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
  },
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
  },
  "UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
  }
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
]
},
],
"rules": [
  {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    {
        "type": "endpoint"
    },
    {
        "type": "tree"
    },
    {
        "conditions": [
            {

```



```

    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsFIPS"
        ]
      }
    ],
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    },
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },

```


both",

```
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
},
{
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
```

```

        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
},
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  },
  {
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  },
  {
    "type": "tree"
  },
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  },
  {
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
]

```

```

    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",

      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

```

```

    "expect": {

"endpoint": {
    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    },
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
    },

```

```

        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {

```

```

        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {

```

```

        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```



```

        {
            "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                }
            },
            "params": {
                "UseFIPS": false,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips enabled and dualstack disabled",
            "expect": {
                "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "For custom endpoint with fips disabled and dualstack enabled",
            "expect": {
                "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": true,
                "Endpoint": "https://example.com"
            }
        },
        {
            "documentation": "Missing region",
            "expect": {
                "error": "Invalid Configuration: Missing Region"
            }
        }
    ],
    "version": "1.0"
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
      "ProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
          "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
        }
      },
      "HealthStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
        "traits": {
          "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
        }
      },
      "Ipv4Address": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
        }
      },
      "Ipv6Address": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "ServerType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  }
}

```

```

    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",

```

```

        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoints": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits":
{
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
    ],

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListProductSubscriptions"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "ProductUserSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
      }
    }
  }
}

```

```

    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        }
    ]
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    }
},
    "traits": {
        "smithy.api#input": {}
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
        "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Product": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {

```

```

    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
      "type": "union",

      "members": {
        "RdsSalSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RDS_SAL",
              "value": "RDS_SAL",
              "documentation": ""
            }
          ]
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },

```



```

    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
    }
  },
  "uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  },

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  },

```



```

"traits": {
  "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    }
  }
}

```

```

    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

```

RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

```

```

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}

```



```

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {

```

```

    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever

```

OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${ github.base_ref }
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit
 path: scancode-toolkit
 fetch-depth: 1
 - name: Set up Python
 \${{ matrix.python-version }}
 uses: actions/setup-python@v2
 with:
 python-version: \${ matrix.python-version }
 # ScanCode

```

- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint

OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```



```

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
            ],
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own

```

```

License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
    "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target":
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            }
        }
    }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {

```



```

        "uri": "/subscription/GetServiceSettings"

    },
    "smithy.api#idempotent": {}
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-

```

```

subscriptions-usage-operation.html\>Usage\n    operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    },
    "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\>Usage\n    operation values</a> in the <i>License Manager User Guide</i> .</p>"
        }
    },
    "LastUpdatedTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
        }
    },
    "SubscriptionName": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
        }
    },
    "OsVersion": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
        }
    },
    "SubscriptionProviderCreateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
        }
    },
    "SubscriptionProviderUpdateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
        }
    },
    "DualSubscription": {
        "target": "smithy.api#String",

```

```
"traits": {
    "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for\n\t\tthe same software on your instance.</p>"
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\tthird-party Linux subscription provider that you've registered with License Manager.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux\nsubscriptions.</p>"
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    }
  ],
```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [

```

```

        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        }
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ]
}

],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    }
}

```

```

        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        }
                                    ],

```



```

        true
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}

],

"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            }
                        ]
                    }
                ]
            },
            {
                "fn": "supportsFIPS"
            }
        ]
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],

"rules": [

      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],

    "type": "tree"
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"

```

```

    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                  ]
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

        }
    },
    "type": "tree"
}

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},

```

```

"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  }
}

```

```

    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"

```



```

    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        }
    },
    {
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {

```

```

        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {

```



```

<code>Status</code>\n          </p>\n          </li>\n          <li>\n          <p>\n          <code>UsageOperation</code>\n          </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use one of the following <code>Operator</code> values to \n \t\ndefine the behavior of the filter:</p>\n        <ul>\n        <li>\n        <p>\n        <code>contains</code>\n        </p>\n        </li>\n        <li>\n        <p>\n        <code>>equals</code>\n        </p>\n        </li>\n        <li>\n        <p>\n        <code>Notequal</code>\n        </p>\n        </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those

```

that match one or more sets of key-value pairs that you specify. For example, you can filter by the name of `Subscription` with an optional operator to see subscriptions that match, partially match, or don't match a certain subscription's name.

The valid names for this filter are:

- `Subscription`

The valid Operators for this filter are:

- `contains`
- `equals`
- `NotEqual`

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  }
}

```



```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {

```

```

        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      }
    },
  },

```

```

"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:"
      }
    }
  }
}

```

```

9-\\.{1,510}/[a-z0-9-\\.{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {

```



```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
    },
    "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
    "type": "map",
    "key":
{
    "target": "smithy.api#String"
},
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested

```

```

\n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  }
}

```



```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)
```

```
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.

```

```

//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}

```



```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}

```

```

}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.71 aws-sdk-go-v2-service-internal-accept-encoding 1.13.3

1.71.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)

```

```

if err != nil {
    return nil, err
}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```



```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.

```

License *types.License

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack, options); err != nil {

return err

}

if err = addClientRequestID(stack); err

!= nil {

return err

}

if err = addComputeContentLength(stack); err != nil {

return err

}

if err = addResolveEndpointMiddleware(stack, options); err != nil {

return err

}

if err = addComputePayloadSHA256(stack); err != nil {

return err

}

if err = addRetry(stack, options); err != nil {

return err

}

if err = addRawResponseToMetadata(stack); err != nil {

return err

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)

```

```

if err != nil {
    return nil, err
}

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

```

```

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```



```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint

```

OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {

```

```

    params = &DeleteLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}

```

```

}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {

```



```

    "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",

            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        },
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
        ],
    },

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },

    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        },
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
      }
    },

    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
},
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate

```

from the EC2 instance.</p>"

```
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  }
}
```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    },
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      }
    },
    "FailureMessage": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
  },

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    }
  }

```

```

    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",

```

```

    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}):|([0-9a-fA-F]{1,4}):{1,6}:[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [

```

```

        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [

```

```

        {
            "ref": "Endpoint"
        }
    ]
}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    }
]
}

```

```

        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                }
            ]
        }
    ]
}

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
    ],
    },
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            }
        ]
    }
]
}

```

```

        ]
    }
},
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

```

```

    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",

      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{

```

```

        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
    },

```

```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {

```



```

        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {

```

```

"type": "structure",
"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of LicenseServer resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      }
    ]
  }
}

```

```

        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken

```


from a previously truncated response.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      }
    ]
  }
}
```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
      "type": "structure",

      "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          }
        ]
      }
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
},
"errors": [
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",

```



```

    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "MaxResults": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    }
  }
}

```

```

    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  }

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
            "smithy.api#required": {}
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        },
        "Settings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
            "traits": {
                "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        },
        "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
            "type": "structure",
            "members": {
                "IdentityProviderSummary": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                    "traits": {
                        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                        "smithy.api#required": {}
                    }
                }
            }
        },

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {

```



```

        "min": 5,
        "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "RDS_SAL",
                "value": "RDS_SAL",
                "documentation": ""
            }
        ]
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input":

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  }
},
"uri": "/user/StartProductSubscription"
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {

```

```

"Username": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
        }
    }
}

```

```

    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {

```

```

    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  }
},

```



```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "TagKeys": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},

```

```

    "traits": {
      "smithy.api#output": {}
    },
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
    }
  }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
      "smithy.api#required": {}
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
      }
    }
  }
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```

```

Serialize stack step
  setOperationInput
  ResolveEndpoint
  OperationSerializer
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  Signing
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  OperationDeserializer
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
  "context"
  "fmt"
  awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
  "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
  "github.com/aws/smithy-go/middleware"
  smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.

```

```

func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

```

```

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

```



```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2

```

with:
  repository: nexB/scancode-toolkit
  path: scancode-toolkit
  fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming

RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step


```

setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns

```

```

...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type ListReceivedLicensesInput struct {

```

```

    // Filters to scope the results. The following filters are supported:

```

```

    //

```

```

    // - ProductSKU

```

```

    //

```

```

    // - Status

```

```

    //

```

```

    //

```

```

    // - Fingerprint

```

```

    //

```

```

    // - IssuerName

```

```

    //

```

```

    // - Beneficiary

```

```

    Filters []types.Filter

```

```

    // Amazon Resource Names (ARNs) of the licenses.

```

```

    LicenseArns []string

```

```

    // Maximum number of results to return in a single call.

```

```

    MaxResults *int32

```

```

    // Token for the next set of results.

```

```

    NextToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type ListReceivedLicensesOutput struct {

```

```

    // Received license details.

```

```

    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return

```

```

err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

    },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The type of name to filter by.</p>"
    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
        },
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
                "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified in the request.</p>"
            }
        }
    }
}

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of the instance.</p>"
            }
        }
    }
},

```

```

"Region": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
  }
},
"UsageOperation": {
  "target": "smithy.api#String",

  "traits": {
    "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n    <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n    operation values</a> in the <i>License Manager User
Guide</i>.</p>"
  }
},
"ProductCode": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
  "traits": {
    "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n    operation values</a> in the <i>License Manager User Guide</i> .</p>"
  }
},
"LastUpdatedTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
  }
},
"SubscriptionName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
  }
},
"OsVersion": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
  }
},
"SubscriptionProviderCreateTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp when you registered

```

```

the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",
        "arnNamespace": "license-manager-linux-subscriptions",
        "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
      },
      "aws.auth#sigv4": {
        "name": "license-manager-linux-subscriptions"
      }
    }
  }
}

```

```

},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,

```

```

        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

```



```

    }

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "endpoint": {
      "url": {
        "ref": "Endpoint"
      },
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    }
  ],

```



```

        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [

                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

"type": "tree"
        },
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{

```

```

"conditions": [
    {
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        "type": "tree"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
],
"rules": [

```

```

        {
            "conditions": [],
            "rules": [

{
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"

```

```

        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,

```



```

        "UseDualStack": false
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        }
    }
}

```

```

    }

  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
}

```

```

    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    }
  }
}

```

```

    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},

{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
}

```

```

    },
    {
      "documentation": "Missing region",

      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Instances"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
    "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
    }
},
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
            "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {

```

```

    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your organization, the returned results will include data aggregated across your accounts in Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",

```



```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>equals</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>Notequal</code>\n            </p>\n        </li>\n    </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {

```

```

"type": "structure",
"members": {
  "Subscriptions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    }
  },

```

```

    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSources": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
}
},
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
    "type": "structure",
    "members": {
      "RegisteredSubscriptionProviders": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",

```



```

    "type": "structure",
    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",

```

```

        "documentation": "Contains operator"
    }
},
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {

```



```

    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#length": {
    "min": 1,
    "max": 100
  },
  "smithy.api#uniqueItems": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {

```

"smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

```

    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",

    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        }
      ]
    }
  }
}

```

```

    },
    {
      "name": "PENDING",
      "value": "PENDING",
      "documentation": "PENDING status"
    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
    "smithy.api#http": {
      "method": "PUT",

```



```

        "min": 0,
        "max": 50
    },
    "smithy.api#sensitive": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  },
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
{}
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\ integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {

```

```

"LinuxSubscriptionsDiscovery": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
  "traits": {
    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
  }
},
"LinuxSubscriptionsDiscoverySettings": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
  "traits": {
    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
  }
},
"Status": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
  }
},
"StatusMessage": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
  "traits": {
    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
  }
},
"HomeRegions": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
  "traits": {
    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
    "smithy.api#error": "client"
  }
}

```

```

    }
  }
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {

```

```

if params == nil {
    params = &DisassociateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DisassociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptors(stack, options); err != nil {

```

```

    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {
```

```

    params = &ListLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {

```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,

```

```

options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {

```

```

    return nil, err
}

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

```



```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.

```

```

//
// This member is required.
LicenseConsumptionToken *string

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {

```

```

return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}

```

```

}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.72 github.com/aws/smithy-go 1.23.2

1.72.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.73 aws-sdk-go-v2-aws-protocol-eventstream 1.7.3

1.73.1 Available under license :

```
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks out the specified license for offline use.
```

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

```

```

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

```



```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

```

```

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint

```

OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step


```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {

```

```

return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {

```

```

    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",

```

```

}
}
CheckoutBorrowLicense
Initialize stack step
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
Serialize stack step
  setOperationInput
  ResolveEndpoint
  OperationSerializer
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  Signing
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  OperationDeserializer
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {

```

```

"type": "structure",
"members": {
  "DomainName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
    }
  },
  "DomainIpv4List": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainIpv6List": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
          }
        },
        {
          "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
              "smithy.api#required": {}
            }
          },
          {
            "Tags": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
              "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
              }
            }
          },
          {
            "traits": {
              "smithy.api#input": {}
            }
          }
        },
        {
          "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
            "type": "structure",
            "members": {
              {
                "IdentityProviderArn": {
                  "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                  "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
                  }
                },
                {
                  "LicenseServerEndpointArn": {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",

```

```

"members": {
  "LicenseServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License

```

```

Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",

```



```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

"target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },

  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Attribute": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Operation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
    }
  },
  "Value": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Value of the filter.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
      "smithy.api#required": {}
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      },
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      },
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      },
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      },
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      },
    }
  }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {

```

```

    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {

```



```

    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}):[0-9a-fA-F]{1,4})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  }

```

```

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
  },
  {

    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",

```

```

    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
      },
      "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
      },
      "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
      }
    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ]
  },
  "rules": [
    {
      "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        },
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        },
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {

```

```

    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsFIPS"
        ]
      }
    ]
  },

  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

  "type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
        "type": "error"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},

```



```

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}
]
}

```

```

        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }

            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},

```

```

"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {

```

```

"endpoint": {
    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
},
"params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                },
                "params": {
                    "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.gov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.gov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack

```

```

disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},

```



```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    }
  },

```

```

"ServerType": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
  "traits": {
    "smithy.api#documentation": "<p>The type of license server.</p>"
  }
},
"ServerEndpoint": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
  "traits": {
    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
  }
},
"LicenseServerEndpointId": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
  "traits": {
    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
  }
},
"LicenseServerEndpointArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
  }
},
"LicenseServerEndpointProvisioningStatus": {
  "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
  "traits": {
    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
  }
},
"LicenseServers": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList",
  "traits": {
    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
  }
},

```

```

    "CreationTime": {
        "target": "smithy.api#Timestamp",

    "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
        "type": "string"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "PROVISIONING",
                    "value": "PROVISIONING",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONING_FAILED",
                    "value": "PROVISIONING_FAILED",
                    "documentation": ""
                },
                {
                    "name": "PROVISIONED",
                    "value": "PROVISIONED",
                    "documentation": ""
                },
                {
                    "name": "DELETING",
                    "value": "DELETING",
                    "documentation": ""
                }
            ],
        }
    },

```

```

        {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
        },
        {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
        }
    ]
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
    }
},
"ServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
    }
}

},
"traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
}

```



```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of InstanceSummary resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                "smithy.api#range": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {

```

"smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },
```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    }
  },

```

```

"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"MaxResults": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
  }
},
"Filters": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
  "traits": {
    "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
  }
}
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next

```

request to retrieve additional objects.</p>"

```
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```



```

        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product

```

```

user.</p>",
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a product for this user.</p>",
      "smithy.api#required": {}
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The end date of a subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  }
},

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/RegisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
    }

},
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
      "type": "structure",
      "members": {
        "SecretId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
            "smithy.api#length": {
              "min": 1
            }
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
      "type": "string",
      "traits": {
        "smithy.api#length": {
          "min": 5,
          "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
      "type": "structure",
      "members": {
        "Endpoint": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
        "type": "union",

        "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RDS_SAL",
                        "value": "RDS_SAL",
                        "documentation": ""
                    }
                ]
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {

```

```

    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/StopProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    }
  }
}

```

```

    },
    "IdentityProvider":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {

```

```

    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  }
}

```



```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "UpdateSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
      "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",

```

```

        "smithy.api#required": {}
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.

```

```

//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}

```



```

}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:

licensescan:

name: License Scan
runs-on: ubuntu-latest

strategy:

matrix:
python-version: [3.9]

steps:

- name: Checkout target
uses: actions/checkout@v2
with:
path: sdkbase
ref: \${ github.base_ref }

- name: Checkout this ref
uses: actions/checkout@v2
with:

path: new-ref
fetch-depth: 0

- name: Get Diff
run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files
run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode
uses: actions/checkout@v2

with:
repository: nexB/scancode-toolkit
path: scancode-toolkit
fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }
uses: actions/setup-python@v2
with:
python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

```

working-directory: ./scancode-toolkit
run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint

OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```



```

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
            ],
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own

```

```

License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
    "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target":
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            }
        }
    }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {

```



```

    "uri": "/subscription/GetServiceSettings"

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-

```

```

subscriptions-usage-operation.html\>Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\>Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",

```



```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [

```

```

        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        }
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ]
}

],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    }
}

```

```

        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

                        {
                            "fn": "aws.partition",
                            "argv": [
                                {
                                    "ref": "Region"
                                }
                            ],
                            "assign": "PartitionResult"
                        }
                    ],
                    "rules": [
                        {
                            "conditions": [
                                {
                                    "fn": "booleanEquals",
                                    "argv": [
                                        {
                                            "ref": "UseFIPS"
                                        }
                                    ],

```

```

        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ]
}

```



```

        }
    ],
    "rules": [
        {
            "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"

```

```

    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                  ]
                }
              ]
            }
          ],
          "rules": [
            {
              "conditions": [],
              "rules": [
                {
                  "conditions": [],
                  "endpoint": {
                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                  },
                  "type": "endpoint"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

        }
    ],
    "type": "tree"
}

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},

```

```

"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    }
},
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"

```

```

    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```



```

    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        }
    },
    {
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {

```

```

        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {

```

```

"type": "string",
"traits": {
  "smithy.api#enum": [
    {
      "name": "Enabled",
      "value": "Enabled",
      "documentation": "Enabled LinuxSubscriptionsDiscovery"
    },
    {
      "name": "Disabled",
      "value": "Disabled",
      "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Instances"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
    <p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
    <ul>\n
      <li>\n
        <p>\n
          <code>AccountID</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>AmiID</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>DualSubscription</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>InstanceID</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>InstanceType</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>ProductCode</code>\n
        </p>\n
      </li>\n
      <li>\n
        <p>\n
          <code>Region</code>\n
        </p>\n
      </li>\n
    </ul>
      "
      }
    }
  }
}

```

```

<code>Status</code>\n          </p>\n          </li>\n          <li>\n          <p>\n          <code>UsageOperation</code>\n          </p>\n          </li>\n        </ul>\n        <p>For each filter, you can use one of the following <code>Operator</code> values to \n \t\define the behavior of the filter:</p>\n        <ul>\n        <li>\n        <p>\n        <code>contains</code>\n        </p>\n        </li>\n        <li>\n        <p>\n        <code>>equals</code>\n        </p>\n        </li>\n        <li>\n        <p>\n        <code>Notequal</code>\n        </p>\n        </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those

```

that match one or more sets of key-value pairs that you specify. For example, you can filter by the name of `Subscription` with an optional operator to see subscriptions that match, partially match, or don't match a certain subscription's name.

The valid names for this filter are:

- `Subscription`

The valid Operators for this filter are:

- `contains`
- `equals`
- `NotEqual`

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
      }
    }
  },

```

```

"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",

```



```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:"
      }
    }
  }
}

```

```

9-\\.[1,510]/[a-z0-9-\\.[1,510}]$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {

```



```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
    }
},
"tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
    "type": "map",
    "key":
{
    "target": "smithy.api#String"
},
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested

```

```

\n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  }
}

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {
```

```
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)
```

```
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.

```

```

//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}

```



```

}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.74 smithy-go 1.23.2

1.74.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.75 github.com/aws/aws-sdk-go-v2/service/internal/accept-encoding 1.13.3

1.75.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
```

```
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)
```

```
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
```

```

//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.

```

```

NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {

```

```

    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}

if err = addClientRequestID(stack); err
!= nil {
    return err
}

if err = addComputeContentLength(stack); err != nil {
    return err
}

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}

if err = addRetry(stack, options); err != nil {
    return err
}

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}

if err = addRecordResponseTiming(stack); err != nil {
    return err
}

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}

if err = addClientUserAgent(stack, options); err != nil {
    return err
}

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

if err = addTimeOffsetBuild(stack, c); err != nil {

```

```

    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```



```

    OperationName: "GetLicense",
  }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //

```

```

// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

```

```

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {

```

```

    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {

```



```

return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
return err
}
}

```

```

if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint

```

OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
 "type": "structure",

```

"members": {
  "DirectoryId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
    "traits": {
      "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv4List",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
            "min": 1,
            "max": 2
        }
    },
    "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 2
            }
        }
    },
    "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
    },
    "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                    "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                }
            ],
        },
    },

```

```

        {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
        }
    ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],

```

```

    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    }
  }
}

```

```

    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    }
  }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of

```

Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.

```
    "smithy.api#required": {}
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The LicenseServerSettings resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the LicenseServerEndpoint
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
```

```

    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ]
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
      "type": "structure",
      "members": {
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>  
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",

```

```

    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {

```

```

    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {

```



```

        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
    }
}
},

"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {

```

```

    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC

```

endpoints.</p>",

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  }
}
```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            },
            "LastStatusCheckDate": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The date of the last status check.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
                }
            },
            "OwnerAccountId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {

```

```

        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
        }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}){1,7}|([0-9a-fA-F]{1,4}):{1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        }
      ]
    }
  }

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],

```



```

"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",

```

```

        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]
],

```

```

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
    },
    "rules": [
        {
            "conditions": [

```

```

    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            }
          ],
          true
        }
      ],
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
              "supportsFIPS"
            }
          ]
        }
      ]
    }
  ]
}

```

```

        ]
      }
    ]
  },

  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    {
      "ref": "UseFIPS"
    },
    true
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",

```



```

        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },

```



```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

```

```

        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
    },

```

```

        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },

```

```

        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    }
  },

```

```

    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license
server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The

```

```

<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {

```



```

    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",
          "documentation": ""
        },
        {
          "name": "DELETED",
          "value": "DELETED",
          "documentation": ""
        }
      ]
    }
  }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      },
      {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
}
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
},
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        }
    },

```

```

    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,

```

```

        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that

```

contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```



```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single

```

```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        }
    }
}

```

```

    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    }
  },

```



```

    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {

```

```

    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
        "type": "union",

        "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {

```

```

    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
    "smithy.api#length": {
      "min": 1
    },
    "smithy.api#required": {}
  }
},
"SecurityGroupId": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
  "traits": {
    "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },

    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
    },
    "IdentityProvider":
{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
    "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnet": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {}
    }
}

```

```

        "min": 0,
        "max": 50
    },
    "smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",

```

```

"members": {
  "ResourceArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
      "smithy.api#httpLabel": {},
      "smithy.api#required": {}
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered
identity provider.</p>",
    "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
        }
    },
    "UpdateSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    }
}

```

```

    }

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    }
}

```



```

    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

```

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //

```

```

// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

```

```

//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {

```

```

return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
    = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

```
}
CheckInLicense
Initialize stack step
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
Serialize stack step
  setOperationInput
  ResolveEndpoint
  OperationSerializer
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  Signing
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  OperationDeserializer
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
  contents: read
```

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test

```



```
run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
AssociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU

```

```

//
// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,

```



```

ServiceID: ServiceID,
OperationName: "ListReceivedLicenses",
}
}
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {

```

```

    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see<n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage<n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage<n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```

```

    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",

```

```

"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }
},
{
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
},
{
    "type": "tree"
}
],
}
],

```

```

    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"

```

```

        },
        true
    ]
}

],

"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        }
                    ]
                }
            ],
            "supportsFIPS"
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [],
            "rules": [
                {

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
}
],
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    }
},
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    }
},
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },

```



```

        {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\define the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
        match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
        <li>\n  <p>\n  <code>Subscription</code>\n  </p>\n  </li>\n  </ul>\n
        <p>The valid Operators for this filter are:</p>\n  <ul>\n  <li>\n  <p>\n  <code>contains</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  <code>equals</code>\n  </p>\n  </li>\n  <li>\n  <p>\n
      
```

```

<code>Notequal</code>\n          </p>\n          </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSources": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {

```

```

"name": "Disabled",
  "value": "Disabled",
  "documentation": "Disabled OrganizationIntegration"
}
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {

```



```

subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      }
    },
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",

```

```

        "documentation": "InProgress status"
    },
    {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

```

```

        {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
    },

```


"smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are formatted as key value pairs.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",

```

```

        "smithy.api#required":
    {}
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    }
}

```

```

    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

```

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {

```



```

return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
return err
}
}

```

```

if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns, c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```

//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```



```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

```

```

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
}

```

```

if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
        return err
    }
    if err = addRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack,
options); err != nil {
        return err
    }
    return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}

```

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}
}

```

1.76 x-sys 0.38.0

1.76.1 Available under license :

```

# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at http://tip.golang.org/CONTRIBUTORS.
Copyright (c) 2009 The Go Authors. All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.77 common 0.67.3

1.77.1 Available under license :

Common libraries shared by Prometheus Go components.

Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.78 prometheus-common 0.67.3

1.78.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.79 aws-sdk-go-v2-service-s3 1.91.1

1.79.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licenseseanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licenseseanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //
```

```

the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```



```

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

//

```

```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```


ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},

```



```

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},

"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```



```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```



```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```


both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
},
],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```



```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
        }
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                "smithy.api#range": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
      "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#readonly": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```



```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```



```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```



```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.

```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```



```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
              "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
          }
        }
      }
    }
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</p> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```



```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        },
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```



```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

```

```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {

"Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "Missing region",

  "expect": {
    "error": "Invalid Configuration: Missing Region"
  }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following <code>Operator</code> values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>Notequal</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  },\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```



```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",

```



```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```



```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.80 github.com/aws/aws-sdk-go-v2/service/internal/checksum 1.9.5

1.80.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```



```

    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

```

```

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```



```

    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            }
        ]
    },

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```



```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```



```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
      }
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```



```

        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            }
          ]
        }
      ],
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```



```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  },
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```



```

    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```



```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

```

```

// License status.
Status types.LicenseStatus

```

```

// License version.
Version *string

```

```

// Metadata pertaining to the operation's result.

```


ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},  
middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}  
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addClientRequestID(stack); err != nil {  
    return err  
}  
if err = addComputeContentLength(stack); err != nil {  
    return err  
}  
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addComputePayloadSHA256(stack); err != nil {  
    return err  
}  
if err = addRetry(stack, options); err != nil {  
    return err  
}  
if err = addRawResponseToMetadata(stack); err != nil {  
    return err  
}  
if err = addRecordResponseTiming(stack); err != nil {  
    return err  
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scandcode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scandcode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput


```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```

```

    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  }
}

```



```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "supportsFIPS"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          }
        ]
      }
    ]
  }
]
}

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```



```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify `Region` in the `Name`, with the \n \t`contains` operator to list all subscriptions that match a partial string in the \n \t`Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key \n \tto streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to \n \tdefine the behavior of the filter:

- `contains`
- `equals`
- `NotEqual`

```
{
  "filters": [
    {
      "name": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "The maximum items to return in a request."
      }
    },
    {
      "name": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse",
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse",
      "traits": {
        "smithy.api#documentation": "A token to specify where to start paginating. This \n\tis the\n\tnextToken from a previously truncated response."
      }
    }
  ],
  "traits": {
    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
}
```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```



```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```

```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
      "smithy.api#httpLabel": {},
      "smithy.api#required": {}
    }
  },
  "tagKeys": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```



```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```



```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.81 github.com/aws/aws-sdk-go-v2/credentials 1.19.0

1.81.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)
```

```
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
```

```
    //
```

```
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
```

```
    //
```

```
    // This member is required.
```

```
    DigitalSignatureMethod types.DigitalSignatureMethod
```

```
    // License entitlements. Partial checkouts are not supported.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.EntitlementData
```

```
    // Amazon Resource Name (ARN) of the license. The license must use the borrow
```

```
    // consumption configuration.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseArn *string
```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)

```



```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license.

```

//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

```

```

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

```



```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```


disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
        "DomainName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```



```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
{
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
{
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```

```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```



```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```

```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```



```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },
    },
    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```

```

        "ref": "UseFIPS"
    },

true

    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
    ]
    }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

```

```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
},
"params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```

```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {

```

```

    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

"type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},
"SubscriptionEndDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```



```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  },
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  },
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```



```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```

```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```

```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```

```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.

```


ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.

//

// This member is required.

Entitlements []types.Entitlement

// Home Region for the license.

//

// This member is required.

HomeRegion *string

// License issuer.

//

// This member is required.

Issuer *types.Issuer

// License name.

//

// This member is required.

LicenseName *string

// Product name.

//

// This member is required.

ProductName *string

// Product SKU.

//

// This member is required.

ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

//

// This member is required.

Validity *types.DatetimeRange

// Information about

the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in

// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde

```

}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.

```

```

MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {

```

```

    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"

```



```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's  
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {

```



```

    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      }
    }
  }

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {

```



```

"type": "structure",
"members": {
  "message": {
    "target": "smithy.api#String"
  }
},
"traits": {
  "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
  "smithy.api#error": "server"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
          "type": "boolean"
        },
        "UseFIPS": {

```



```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"

```

```

    },
    "supportsFIPS"
  ]
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [

        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],

```

```

        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {

```



```

        },
        "type": "endpoint"
    }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            }
        }
    ]
}
}

```



```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```



```

    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
        "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Notequal</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```



```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```



```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.

```

Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.82 github.com/aws/aws-sdk-go-v2/internal/configsources 1.4.14

1.82.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```



```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```



```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```

```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```



```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/user/AssociateUser"
},
"smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
    },
  },

```



```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The LicenseServerSettings resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the LicenseServerEndpoint
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

    "uri": "/license-server/DeleteLicenseServerEndpoint"

  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```


"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        },
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        },
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        },
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        },
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        },
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```



```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
        true
    },
    {
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ],
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                        ],
                        "supportsFIPS"
                    },
                ],
            },
        ],
    },
],
},

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
},
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
  },
  true
]
}
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [

              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```



```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.gov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
      "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
    {
        "type": "string",
        "traits": {
            "smithy.api#length": {
                "min": 5,
                "max": 200
            },
            "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
        "type": "structure",
        "members": {
            "Endpoint": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```



```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
    ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions


```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License status.

Status types.LicenseStatus

// License version.

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

CheckoutLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```



```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

    "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":

```

```

    "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
      }
    }
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
      },
      true
    ]
  }
},
{
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
{
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
{
  "type": "tree"
},
{
  "type": "tree"
},
{

```

```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        ]
      }
    ]
  }
]

```

```

    }

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ],
        "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```

```

        "supportsFIPS"
    ]
},
true

]
}
],
"rules": [
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},

                "headers": {}
            },
            "type": "endpoint"
        ]
    },
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
            true

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```



```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\nA token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\n<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional\nobjects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSources": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

        "traits": {
            "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```



```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]^{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```

```

        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```

```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//

```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```



```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
}
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```
    //
```

```
    // - Fingerprint
```

```
    //
```

```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```



```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```



```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.83 github.com/aws/aws-sdk-go-v2/internal/endpoints/v2 2.7.14

1.83.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//
the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptors(stack, options); err != nil {
return err
}

```



```

    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

```

```

//
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```



```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```



```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},

"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```



```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:([0-9a-fA-F]{1,4}:|::|([0-9a-fA-F]{1,4}:|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:|0-9a-fA-F){1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
  },

```



```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```



```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS":
false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint":
{
        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS":
false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```



```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err
= addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help


```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```



```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            {
              "fn": "supportsDualStack"
            }
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        },
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                    },
                    "properties": {},
                    "headers": {}
                },
                {
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```



```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    }
  }
}

```

```

    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {

"Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "Missing region",

  "expect": {
    "error": "Invalid Configuration: Missing Region"
  }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {

```



```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>Notequal</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  },\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
"name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```



```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
}
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```



```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```



```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.84 aws-sdk-go-v2-service-internal-checksum 1.9.5

1.84.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```



```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

```

```

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

```



```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```



```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```



```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
  "smithy.api#error": "server",
  "smithy.api#httpError": 500
}
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {

```



```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                },
                "assign": "PartitionResult"
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                    ],
                }
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
        }
    ]
},
    true

```

```

        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]

```

```

        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [

```



```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```



```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
        },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```



```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```


setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```



```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  }
}

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          }
        ]
      }
    ]
  }
]
}

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```



```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```



```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria.

For example, you can specify `Region` in the `Name`, with the `contains` operator to list all subscriptions that match a partial string in the `Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key to streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to define the behavior of the filter:

- `contains`
- `equals`
- `Notequal`

```
{
  "filters": [
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "Operator": "contains",
      "Value": "us-west-2"
    },
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse",
      "Operator": "equals",
      "Value": "true"
    }
  ],
  "MaxResults": 10,
  "NextToken": null
}
```

The maximum items to return in a request.

A token to specify where to start paginating. This is the nextToken from a previously truncated response.

smithy.api#length: {
 min: 1,
 max: 16384
}

traits: {
 smithy.api#documentation:
 NextToken length limit is half of ddb accepted limit.
 Increase this limit if parameters in request increases.
 }

com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse: {
 type: "structure",
 members: {
 Instances: {
 target: "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
 traits: {

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```



```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/UpdateServiceSettings"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
{}
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }  
}
```



```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```



```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```



```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.85 github.com/aws/aws-sdk-go-v2/service/internal/s3shared 1.19.14

1.85.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string
}
```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```



```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license.

```

//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

```

```

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```



```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
        "DomainName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",

```



```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
            "smithy.api#required": {}
        }
    },
    "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
            "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
            "smithy.api#required": {}
        }
    }
},
{
    "traits": {
        "smithy.api#input": {}
    }
},
{
    "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoint": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
                }
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
{
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        }
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```



```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^([a-z0-9-]{10})$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```



```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,6}|[0-9a-fA-F]{1,4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```

```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```

```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },
    },
    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```

```

        "ref": "UseFIPS"
    },

true

    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
    ]
    }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

```



```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
},
"params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```



```

    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```

```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

"type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
]

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},
"SubscriptionEndDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```

```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```



```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```

```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "TagKeys": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```



```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
          }
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```

```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.

```

ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.

//

// This member is required.

Entitlements []types.Entitlement

// Home Region for the license.

//

// This member is required.

HomeRegion *string

// License issuer.

//

// This member is required.

Issuer *types.Issuer

// License name.

//

// This member is required.

LicenseName *string

// Product name.

//

// This member is required.

ProductName *string

// Product SKU.

//

// This member is required.

ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

//

// This member is required.

Validity *types.DatetimeRange

// Information about

the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in

// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde

```

}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```



```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

 steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.

```



```

MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {

```

```

    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's  
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {

```

```
"smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
      }
    },
    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
```

```

    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      }
    }
  }

```



```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
        }
    },
    "SubscriptionProviderCreateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
        }
    },
    "SubscriptionProviderUpdateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
        }
    },
    "DualSubscription": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
        }
    },
    "RegisteredWithSubscriptionProvider": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ]
}

```

```

    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
          "type": "boolean"
        },
        "UseFIPS": {

```

```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"

```



```

    },
    "supportsFIPS"
  ]
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [

        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],

```

```

        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {

```



```

        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
    "type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            }
        }
    ],
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}
]
}
}

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```



```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
        "Instances": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
            "traits": {
                "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```



```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
    "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
  {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```

```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type AssociateLicenseInput struct {

```

```

    // The type of license to associate with the workspace.

```

```

    //

```

```

    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.

```

```

    //

```

```

    // This member is required.

```

```

    LicenseType types.LicenseType

```

```

    // The ID of the workspace to associate the license with.

```

```

    //

```

```

    // This member is required.

```

```

    WorkspaceId *string

```

```

    // A token from Grafana Labs that ties your Amazon Web Services account with a

```

```

    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

    //

```

```

    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise

```

```

    GrafanaToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type AssociateLicenseOutput

```

```

struct {

```

```

    // A structure containing data about the workspace.

```

```

    //

```

```

    // This member is required.

```

Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```



```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.86 [github.com/aws/aws-sdk-go-](https://github.com/aws/aws-sdk-go-v2/feature/s3/manager) v2/feature/s3/manager 1.20.9

1.86.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```



```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```

```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```

```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
 "type": "structure",
 "members": {
 "DirectoryId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
 "traits": {
 "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```



```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#Ipv6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/user/AssociateUser"
},
"smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
    },
  },
},

```

```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```

"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    ],
    "assign": "PartitionResult"
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseDualStack"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              }
            ]
          }
        ]
      }
    ]
  },

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
  },
  true
]
}
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [

              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
},
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```



```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.gov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```



```

    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
}
},

```

```

    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",

```



```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```



```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
          },
          {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```



```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

CheckoutLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions


```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```

```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

    "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":

```

```

    "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
    "smithy.api#required": {}
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
        }
      }
    }
  },

```



```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
      },
      true
    ]
  }
},
{
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
}
},
{
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```



```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        ]
      }
    ]
  }
]

```

```

    }

    ],

    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                }
                            ]
                        }
                    ]
                },
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                }
                            ]
                        }
                    ]
                }
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        {
                            "ref": "supportsDualStack"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-

```

```

"rules": [

```

```

    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```



```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",

```



```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\nA token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\n<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional\nobjects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n    <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>Subscription</code>\n        </p>\n      </li>\n    </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>"
        }
      }
    },
    "traits": {

```



```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSources": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

        "traits": {
          "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {}
      }
    }
  }
}

```



```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```



```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//

```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

// Filters to scope the results. The following filters are supported:

//

// - Beneficiary

//

// - ProductSKU

//

// - Fingerprint

//

```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```



```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.87 aws-sdk-go-v2-feature-s3-manager

1.20.9

1.87.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//
the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {

```



```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptors(stack, options); err != nil {
return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

```

```

//
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```



```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"


```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```



```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:([0-9a-fA-F]{1,4}:|::|([0-9a-fA-F]{1,4}:|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:|0-9a-fA-F){1,4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```



```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```



```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```



```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                "smithy.api#range": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```



```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```



```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see<n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage<n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```

\\n\\t\\tthe same software on your instance.</p>"

```
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \\n\\t\\t\\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```



```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        },
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  }
]

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```



```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

```

```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
}

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \\\n\\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  }\n},\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \\\n\\tfield\nisn't empty, there are additional elements that the service hasn't \\\n\\tincluded in this request. Use this token with the\nnext request to retrieve \\\n\\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```



```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```



```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.88 aws-sdk-go-v2-internal-configsources

1.4.14

1.88.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```



```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

```



```

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```



```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```



```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```



```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      },
      "com.amazonaws.licenseanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
        "type": "string",
        "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}:|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4:}){1,7}:|([0-9a-fA-F]{1,4:}){1,6}:|0-9a-fA-F]{1,4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```



```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```



```

    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```



```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  }
},
"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```



```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

```

```

// License status.
Status types.LicenseStatus

```

```

// License version.
Version *string

```

```

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},  
middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}  
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addClientRequestID(stack); err != nil {  
    return err  
}  
if err = addComputeContentLength(stack); err != nil {  
    return err  
}  
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addComputePayloadSHA256(stack); err != nil {  
    return err  
}  
if err = addRetry(stack, options); err != nil {  
    return err  
}  
if err = addRawResponseToMetadata(stack); err != nil {  
    return err  
}  
if err = addRecordResponseTiming(stack); err != nil {  
    return err  
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scandcode-toolkit
    fetch-depth: 1
  - name: Set up Python
    ${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scandcode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
    toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
    toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
    are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```



```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```



```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  }
}

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```



```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```



```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```


that you can use to filter the results by your specified criteria.

For example, you can specify `Region` in the `Name`, with the `contains` operator to list all subscriptions that match a partial string in the `Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key to streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to define the behavior of the filter:

- `contains`
- `equals`
- `Notequal`

```
{
  "filters": [
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "Operator": "contains",
      "Value": "us-west-2"
    },
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse",
      "Operator": "equals",
      "Value": "true"
    }
  ],
  "MaxResults": 10,
  "NextToken": null
}
```

The maximum items to return in a request.

A token to specify where to start paginating. This is the nextToken from a previously truncated response.

smithy.api#length: {
 min: 1,
 max: 16384
}

traits: {
 smithy.api#documentation:
 NextToken length limit is half of ddb accepted limit.
 Increase this limit if parameters in request increases.
 }

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
      isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
      next request to retrieve \n\tadditional
      objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t\torganization, the
    returned results will include data aggregated across your accounts in\n\t\tOrganizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```



```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```



```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```



```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.89 github.com/aws/aws-sdk-go-v2/config

1.32.0

1.89.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licenssemanager
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```



```

)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license.


```

//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

```

```

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```



```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            },
            "DomainIpv6List": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
{
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
{
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```

```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  }
},

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {

```



```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```



```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    }
],
{
    {
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    }
},
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```

```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },
    },
    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```

```

        "ref": "UseFIPS"
    },

true

    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
    ]
    }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

```

```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
},
"params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```



```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {

```

```

    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        },
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        },
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        },
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        },
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```


empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
]

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},
"SubscriptionEndDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```

```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```

```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```

```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```



```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.

```

ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.

//

// This member is required.

Entitlements []types.Entitlement

// Home Region for the license.

//

// This member is required.

HomeRegion *string

// License issuer.

//

// This member is required.

Issuer *types.Issuer

// License name.

//

// This member is required.

LicenseName *string

// Product name.

//

// This member is required.

ProductName *string

// Product SKU.

//

// This member is required.

ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

//

// This member is required.

Validity *types.DatetimeRange

// Information about

the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in

// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde

```

}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.

```

```

MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {

```

```

    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {

```



```

    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
                }
            }
        }
    }

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ]
}

```

```

    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
          "type": "boolean"
        },
        "UseFIPS": {

```

```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"

```

```

    },
    "supportsFIPS"
  ]
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [

        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],

```

```

        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {

```



```

        },
        "type": "endpoint"
    }
},
    "type": "tree"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
    "type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            }
        }
    ]
}
}

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```



```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n      Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n      Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      },
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```



```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```



```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```

```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.

```

Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```



```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.90 github.com/aws/aws-sdk-go-v2/internal/v4a 1.4.14

1.90.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```



```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```

```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```



```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
 "type": "structure",
 "members": {
 "DirectoryId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
 "traits": {
 "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/user/AssociateUser"
},
"smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
    },
  },
},

```

```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The LicenseServerSettings resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the LicenseServerEndpoint
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```

"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        },
        "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  },
  {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    }
                ]
            }
        ],
    },

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
    },
    true
]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "fn": "getAttr",
                    "argv": [

{
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
    }
],
"rules": [
{
    "conditions": [],
    "endpoint": {

"url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```



```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS":
true,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.gov.gov"
    }
},
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    }
  },

```



```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```



```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```



```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ]
    }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read


```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

CheckoutLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```

```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

        "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":

```

```

    "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
    "smithy.api#required": {}
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
        }
      }
    }
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
      },
      true
    ]
  }
},
{
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
}
},
{
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        }
      ]
    ]
  }
]

```

```

    }

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ],
        "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```



```

        "supportsFIPS"
      ]
    },
    true

  ]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    ]
  },
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
      ],
      true
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
"params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
}
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```



```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSources": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

        "traits": {
            "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
        }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]^{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```



```

    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```

```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//


```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

// Filters to scope the results. The following filters are supported:

//

// - Beneficiary

//

// - ProductSKU

//

// - Fingerprint

//


```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```



```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.91 aws-sdk-go-v2-service-ssooidc 1.35.8

1.91.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //

```

```

the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)

```



```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

//

```

```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry


```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      }
    }
  },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```



```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```



```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```



```

    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
        ]
    }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```



```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      }
    },
    "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```


}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```



```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```



```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

```



```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  },\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```



```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```



```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.92 aws-sdk-go-v2-credentials 1.19.0

1.92.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```
    out := result.(*CheckoutBorrowLicenseOutput)
```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed

```

[]types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return
```

```
    err
```

```
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
```

```
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
```

```
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (

```



```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
```

```

    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```

```

    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.

```

EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.

LicenseArn

*string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack,

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter

```


SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,

```

```

c.addOperationDeleteLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            }
        ]
    },

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      },
      "com.amazonaws.licenseanagerusersubscriptions#InternalServerException": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
        "type": "string",
        "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                },
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                    ],
                }
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
        }
    ]
}
true

```

```

        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
},
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```



```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]

```

```

        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```



```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    },
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    },
  },
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      },
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      },
    },
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```



```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```



```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```



```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  }
}

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```



```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          }
        ]
      }
    ]
  }
]
}

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria.

For example, you can specify `Region` in the `Name`, with the `contains` operator to list all subscriptions that match a partial string in the `Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key to streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to define the behavior of the filter:

- `contains`
- `equals`
- `Notequal`

```
{
  "filters": [
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "Operator": "contains",
      "Value": "us-west-2"
    },
    {
      "Name": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse",
      "Operator": "equals",
      "Value": "true"
    }
  ],
  "MaxResults": 10,
  "NextToken": null
}
```

The maximum items to return in a request.

A token to specify where to start paginating. This is the nextToken from a previously truncated response.

smithy.api#length: {
 min: 1,
 max: 16384
}

traits: {
 smithy.api#documentation:
 NextToken length limit is half of ddb accepted limit.
 Increase this limit if parameters in request increases.
 }

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ]
  }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```



```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\&#xA0; integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\&#xA0; allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.93 aws-sdk-go-v2-feature-ec2-imds 1.18.14

1.93.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licenssemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```



```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.

```

```

Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the

```

```

// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.

```

```

Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
return err
}

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming

```

```

    RequestResponseLogger
ListLicenses
Initialize stack step
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
Serialize stack step
    setOperationInput
    ResolveEndpoint
    OperationSerializer
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
Finalize stack step
    spanRetryLoop
    InterceptBeforeRetryLoop
    Retry
    InterceptAttempt
    RetryMetricsHeader
    ResolveAuthScheme
    GetIdentity
    ResolveEndpointV2
    disableHTTPS
    ComputePayloadHash
    setLegacyContextSigningOptions
    Signing
Deserialize stack step
    AddRawResponseToMetadata
    ErrorCloseResponseBody
    CloseResponseBody
    ResponseErrorWrapper
    RequestIDRetriever
    OperationDeserializer
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```

```

"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
      "traits": {

```


"smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"

}

},

"IsSharedActiveDirectory": {

"target": "smithy.api#Boolean",

"traits": {

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

}

}

},

"traits": {

"smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"

}

},

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {

"type": "structure",

"members": {

"DomainName": {

"target": "smithy.api#String",

"traits": {

"smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"

}

},

"DomainIpv4List": {

"target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",

"traits": {

"smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",

"smithy.api#length": {

"min": 1,

"max": 2

}

}

},

"DomainIpv6List": {

"target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",

"traits": {

"smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",

"smithy.api#length": {

"min": 1,

"max": 2

}

}

},

```

    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
      }
    }
  }

```

```

    },
    "output": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {

```

```

"Username": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"InstanceId": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
    ]
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
}
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the

```



```

<code>LicenseServerEndpoint</code> resource to delete.</p>",
    "smithy.api#required": {}
  }
},
"ServerType": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
  "traits": {
    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
    "smithy.api#required": {}
  }
}
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            }
        ]
    }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",

```

```

        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",

```

```

"members": {
  "ActiveDirectoryIdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
    },
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    }
  }

```



```

    }
  },
  "Products": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
      "smithy.api#required": {}
    }
  },
  "LastStatusCheckDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date of the last status check.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",

```

```

"members": {
  "Username": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

"target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
            "smithy.api#error": "server",
            "smithy.api#httpError": 500
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
        }
      ],

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",

```

```

        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",

```

```

        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    }
],
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},

```

```

    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        }
      },

      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Region"
        }
      ]
    }
  ]
},
],
"rules": [
  {
    "conditions": [
      {
        "fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      },
    ],
    "assign": "PartitionResult"
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          }
        ],

```


true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    },
    {
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "DualStack is enabled but this partition does not support DualStack",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        }
    }

```



```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```



```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```



```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

"min": 0,
  "max": 50
},
"smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
CreateLicense
Initialize stack step
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
Serialize stack step
  setOperationInput
  ResolveEndpoint
  OperationSerializer
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  Signing
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  OperationDeserializer
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //

```

```

// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

```



```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err
= addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
    err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

 steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref

```

    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming


```

    RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

```

```

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {

```

```

    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            }
        }
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
    ],
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's  
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration  
resource specified in the request.</p>"
      }
    }
  },

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        }
    }
},

```

```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    },
    "LastUpdatedTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
      }
    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {

```

```

    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
      },
      "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the

```

configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```
    "type": "boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
  }
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
```



```

        "ref": "UseDualStack"
    },
    true
]
}

],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "aws.partition",
        "argv": [
            {
                "ref": "Region"
            }
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
],

"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "supportsFIPS"
]

```

```

        ]
    }
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"

```

```

    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "rules": [
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

```

```

"properties": {},

                                "headers": {}
                                },
                                "type": "endpoint"
                                }
                                ],
                                "type": "tree"
                                }
                                ],
                                "type": "tree"
                                },
                                {
                                    "conditions": [],
                                    "error": "FIPS is enabled
but this partition does not support FIPS",
                                    "type": "error"
                                    }
                                ],
                                "type": "tree"
                                },
                                {
                                    "conditions": [
                                        {
                                            "fn": "booleanEquals",
                                            "argv": [
                                                {
                                                    "ref": "UseDualStack"
                                                },
                                                true
                                            ]
                                        }
                                    ]
                                }
                                ],
                                "rules": [
                                    {
                                        "conditions": [
                                            {
                                                "fn": "booleanEquals",
                                                "argv": [
                                                    true,
                                                    {
                                                        "fn": "getAttr",
                                                        "argv": [
                                                            {
                                                                "ref": "PartitionResult"
                                                            },
                                                                "supportsDualStack"
                                                        ]
                                                    }
                                                ]
                                            }
                                        ]
                                    }
                                ]

```

```

        ]
    }
]
}
],
"rules": [
    {
        "conditions": [],
        "rules": [

{
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}

],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ]
}

```

```

        ],
        "type": "tree"
    }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        }
    ],
}

```

```

{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  }
}

```



```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

```

```

{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},

{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
<ul>\n
<li>\n
<p>\n
<code>AccountID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>AmiID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>DualSubscription</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceType</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>ProductCode</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Region</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Status</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>UsageOperation</code>\n
</p>\n
</li>\n
</ul>\n
<p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
<ul>\n
<li>\n
<p>\n
<code>contains</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>equals</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Notequal</code>\n
</p>\n
</li>\n
</ul>"
          }
        },
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation":

```

```

"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\torganization, the returned results will include data aggregated across your accounts in\n\tOrganizations.</p>",
      "smithy.api#http": {
        "code": 200,

```



```

    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n        </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>>equals</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>NotEqual</code>\n            </p>\n            </li>\n        </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  },
},

```

```

    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

```

"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",

```

```

        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License

```


Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"

```
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]^{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",

```

```

"member": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    }
  }
},
"traits": {
  "smithy.api#length": {
    "min": 1,
    "max": 100
  },
  "smithy.api#uniqueItems": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",

    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "RedHat",
                "value": "RedHat",
                "documentation": "RedHat subscription provider namespace"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",

```

```

        "documentation": "ACTIVE status"
    },
    {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
    },
    {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```



```

    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    },
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    }
  ]
}

```



```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {

"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
    "smithy.api#error": "client"
  }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```

)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

```

```

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {

```

```

    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

```

```

}

result, metadata, err := c.invokeOperation(ctx,
"AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```



```

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```

```

if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.94 aws-sdk-go-v2 1.40.0

1.94.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package licenssemanager

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

```

```

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```



```

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

    }
    if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
    err != nil {
        return err
    }
    if err = addRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options); err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```

```

"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.

```

```

KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```



```

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:  ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step

```

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Deletes the specified license.

```
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))  
(*DeleteLicenseOutput, error) {  
    if params == nil {  
        params = &DeleteLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,  
    c.addOperationDeleteLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DeleteLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type DeleteLicenseInput struct {

// Amazon Resource Name (ARN) of the license.

//

// This member is required.

LicenseArn *string

// Current version of the license.

//

// This member is required.

SourceVersion

*string

noSmithyDocumentSerde

}

type DeleteLicenseOutput struct {

```

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
}

```

```

if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```



```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {

```

"smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"

```
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
```

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
  },
```

```
  "DomainIpv4List": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
```

```
  "DomainIpv6List": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
    "traits": {
```

```

    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
        "min": 1,
        "max": 2
    }
},
"DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
},
"DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager

```

```

secret that enables administrators to add or remove users associated with the license server.</p>",
    "smithy.api#required": {}
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      }
    },
    "smithy.api#idempotent": {}
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
      "type": "structure",
      "members": {
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    }
  },
  "Product": {
    "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([sd])-([0-9a-f]){10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
        }
      },
      "InstanceId": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  }
}

```



```

    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
        }
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {

"target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",

```

```

    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}:|([0-9a-fA-F]{1,4}:{1,6}:|0-9a-fA-F){1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
],
"additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to

```

the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```
    "type": "boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
  }
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  },
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        }
      ]
    }
  ]
}
```



```

        },
        true
    ]
}
],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },

"properties": {},
    "headers": {}
},
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        }
    },
    "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            }
          ]
        }
      ],
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
]
},
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ]
},
],

```

```

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [

              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {

          "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [

```

```

        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        }
    ]
}

```

```

    }
  },
  "params": {
    "Region": "us-east-1",

    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {

    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":

```

```

true,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {

```



```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

"endpoint": {
    "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
},
},
"params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
}
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {

```

```

        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    },
    "Ipv6Address": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {

```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code> resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",
          "documentation": ""
        },
        {
          "name": "DELETED",
          "value": "DELETED",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "HEALTHY",
          "value": "HEALTHY",

```

```

        "documentation": ""
    },
    {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
    },
    {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits":
    {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },

        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
},
"errors": [
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
}
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",

```

```

    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummaries": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output":
{}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    }
  },
  "traits": {

```



```

    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
        "smithy.api#error": "client",
        "smithy.api#httpError": 404
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
      "type": "structure",
      "members": {
        "SecretId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains

```

```

credentials.</p>",
    "smithy.api#length": {
        "min": 1
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688

```

communication between resources in your VPC and the VPC endpoint for activation servers.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
},
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input":
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
```

to appear for billing periods that haven't closed (marked as **Pending** billing status) in Amazon Web Services Billing. For more information, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html> Viewing your monthly charges in the Amazon Web Services Billing User Guide.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",

"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: VISUAL_STUDIO_ENTERPRISE | VISUAL_STUDIO_PROFESSIONAL | OFFICE_PROFESSIONAL_PLUS | REMOTE_DESKTOP_SERVICES </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
```

```

        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    }
},

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licenseanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
          "target": "smithy.api#String"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {

```



```

    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
    }
},
"Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
    "method": "POST",

```

```

        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
            }
        },
        "UpdateSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
            "traits": {
                "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

```

RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a

```



```

// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html

```

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License status.

Status types.LicenseStatus

// License version.

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack, options); err != nil {

return err

}

if err = addClientRequestID(stack); err != nil {

return err

}

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2

```

with:
  path: sdkbase
  ref: ${ github.base_ref }
- name: Checkout this ref
  uses: actions/checkout@v2
with:
  path: new-ref
  fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${ github.sha } > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
with:
  repository: nexB/scancode-toolkit
  path: scancode-toolkit
  fetch-depth: 1
- name: Set up Python
${ matrix.python-version }
  uses: actions/setup-python@v2
with:
  python-version: ${ matrix.python-version }
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
AssociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step

```

setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent

AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.

```

```

LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
    },
}

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },

```



```

    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists

```

the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"

```
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    }
  }
}
```

```

    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    },
    "LastUpdatedTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
      }
    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        }
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",

```



```

        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],

```

```

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseDualStack"
          },
          true
        ]
      }
    ]
  },
  {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "endpoint": {
      "url": {
        "ref": "Endpoint"
      },
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    }
  ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ],
            "assign": "PartitionResult"
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseFIPS"
                  },
                  true
                ]
              },
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            }
        ],
        "supportsFIPS"
    ]
}
],
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
],
"type": "tree"

```

one or both",

```
    },
    {
      "conditions": [],
      "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ],
      "rules": [
```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},

                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
]
},
],
"rules": [
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-

```

```

subscriptions.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    }
  ],

```



```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
            }
        }
    },
    {
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    },

```

```

{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    }
  },
  {
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```



```

        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },

```

```

    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",

      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
},
"NextToken": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
            "min": 1,
            "max": 16384
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    },
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
    returned results will include data aggregated across your accounts in\n    Organizations.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
                that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
                <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
                match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
                <li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
                <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
                <code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
                <code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
                <code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
                nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,

```

```

        "max": 16384
    }
}
},
"traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "Subscriptions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
            "traits": {
                "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
            }
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
    "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
        }
    }
}
}

```



```

        "value": "Equal",
        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    }
}

```



```

    "traits": {

        "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
                "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
            }
        },
        "SubscriptionProviderArn": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
            }
        },
        "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
        }
    }
}

```

```

    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
      }
    },
    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]\\{0,63\\}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {

```

"smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

```

    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "ACTIVE",
      "value": "ACTIVE",
      "documentation": "ACTIVE status"
    },
    {
      "name": "INVALID",
      "value": "INVALID",
      "documentation": "INVALID status"
    },
    {
      "name": "PENDING",
      "value": "PENDING",
      "documentation": "PENDING status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",

```



```

    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required": true
            }
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
            "smithy.api#required": true
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux

```

subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home Region.

```

    }
  }

},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux subscriptions.</p>"
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```



```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type AssociateLicenseInput struct {

```

```

    // The type of license to associate with the workspace.

```

```

    //

```

```

    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.

```

```

    //

```

```

    // This member is required.

```

```

    LicenseType types.LicenseType

```

```

    // The ID of the workspace to associate the license with.

```

```

    //

```

```

    // This member is required.

```

```

    WorkspaceId *string

```

```

    // A token from Grafana Labs that ties your Amazon Web Services account with a

```

```

    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

    //

```

```

    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise

```

```

    GrafanaToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type AssociateLicenseOutput

```

```

struct {

```

```

    // A structure containing data about the workspace.

```

```

    //

```

```

    // This member is required.

```

Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err  
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err  
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err  
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err  
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err  
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err  
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err  
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err  
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err  
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err  
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.95 aws-aws-sdk-go-v2-service-signin 1.0.1

1.95.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```



```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```



```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```

```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```



```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 },
 "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
 "type": "structure",
 "members": {
 "DirectoryId": {
 "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
 "traits": {
 "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/user/AssociateUser"
},
"smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },

```



```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```


"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        },
        "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```



```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    }
                ]
            }
        ],
    },

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
  },
  true
]
}
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [

              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```



```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
},
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
}
},

```

```

    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```



```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
    ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions


```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
AssociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```



```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

    "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":

```

```

    "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
      }
    }
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
    },
    true
]
}

],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{

```

```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        }
      ]
    ]
  }
]

```

```

    }

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ],
        "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```

```

        "supportsFIPS"
      ]
    },
    true

  ]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
      ],
      true
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```



```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
"params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
}
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceType</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>ProductCode</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Region</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Status</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>UsageOperation</code>\n                </p>\n                </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>contains</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>equals</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Notequal</code>\n                </p>\n                </li>\n        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
        "type": "structure",
        "members": {
            "SubscriptionProviderSources": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

        "traits": {
            "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```



```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]^{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```

```

        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```

```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//

```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```



```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{ }  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```
    //
```

```
    // - Fingerprint
```

```
    //
```

```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```



```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```



```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.96 aws-sdk-go-v2-config 1.32.0

1.96.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //

```

```

the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutBorrowLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```



```

}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

//

```

```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```



```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```



```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```



```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
  },

```



```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```



```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```



```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

    },
    "Settings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],

  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help


```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```



```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
              "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
          }
        }
      }
    }
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```



```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```



```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
          "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
          returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \\\n\\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  },\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \\\n\\tfield\nisn't empty, there are additional elements that the service hasn't \\\n\\tincluded in this request. Use this token with the\nnext request to retrieve \\\n\\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
            "smithy.api#http": {
                "method": "DELETE",
                "uri": "/tags/{resourceArn}"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
        "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```



```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```



```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```



```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.97 github.com/aws/aws-sdk-go-v2/service/signin 1.0.1

1.97.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```



```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

```

```

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```



```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```



```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            }
        ]
    },

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```



```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
      },
    },
    "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {

```



```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}:|::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4:}{1,7}:|([0-9a-fA-F]{1,4:}{1,6}:|0-9a-fA-F){1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                },
                "assign": "PartitionResult"
            ]
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
}

```

```

        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```



```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",

```



```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        }
    ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  }
},
"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```



```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
```

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},  
middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}  
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addClientRequestID(stack); err != nil {  
    return err  
}  
if err = addComputeContentLength(stack); err != nil {  
    return err  
}  
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addComputePayloadSHA256(stack); err != nil {  
    return err  
}  
if err = addRetry(stack, options); err != nil {  
    return err  
}  
if err = addRawResponseToMetadata(stack); err != nil {  
    return err  
}  
if err = addRecordResponseTiming(stack); err != nil {  
    return err  
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```


setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```



```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          }
        ]
      }
    ]
  }
]
}

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```



```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```



```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify

<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n

<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto streamline results:</p>\n

- \n <p>\n <code>AccountID</code>\n \n \n <p>\n <code>AmiID</code>\n \n \n <p>\n <code>DualSubscription</code>\n \n \n <p>\n <code>InstanceID</code>\n \n \n <p>\n <code>InstanceType</code>\n \n \n <p>\n <code>ProductCode</code>\n \n \n <p>\n <code>Region</code>\n \n \n <p>\n <code>Status</code>\n \n \n <p>\n <code>UsageOperation</code>\n \n

<p>For each filter, you can use one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n

- \n <p>\n <code>contains</code>\n \n \n <p>\n <code>equals</code>\n \n \n <p>\n <code>Notequal</code>\n \n

"

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {

```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
      isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
      next request to retrieve \n\tadditional
      objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
    returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ]
  }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```

```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```



```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```



```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```



```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.98 aws-sdk-go-v2-service-sso 1.30.4

1.98.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licenssemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.

```



```

Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the

```

```

// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.

```

```

Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{ }, middleware.After)
if err != nil {
return err
}

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:  ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming

```



```

    RequestResponseLogger
ListLicenses
Initialize stack step
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
Serialize stack step
    setOperationInput
    ResolveEndpoint
    OperationSerializer
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
Finalize stack step
    spanRetryLoop
    InterceptBeforeRetryLoop
    Retry
    InterceptAttempt
    RetryMetricsHeader
    ResolveAuthScheme
    GetIdentity
    ResolveEndpointV2
    disableHTTPS
    ComputePayloadHash
    setLegacyContextSigningOptions
    Signing
Deserialize stack step
    AddRawResponseToMetadata
    ErrorCloseResponseBody
    CloseResponseBody
    ResponseErrorWrapper
    RequestIDRetriever
    OperationDeserializer
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```

```

"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {

```

"smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"

}

},

"IsSharedActiveDirectory": {

"target": "smithy.api#Boolean",

"traits": {

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

}

}

},

"traits": {

"smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"

}

},

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {

"type": "structure",

"members": {

"DomainName": {

"target": "smithy.api#String",

"traits": {

"smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"

}

},

"DomainIpv4List": {

"target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",

"traits": {

"smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",

"smithy.api#length": {

"min": 1,

"max": 2

}

}

},

"DomainIpv6List": {

"target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",

"traits": {

"smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",

"smithy.api#length": {

"min": 1,

"max": 2

}

}

},

```

    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Arn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
      }
    }
  }

```



```

    },
    "output": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {

```

```

"Username": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"InstanceId": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        }
    ]
},

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
}
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the

```

```

<code>LicenseServerEndpoint</code> resource to delete.</p>",
    "smithy.api#required": {}
  }
},
"ServerType": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
  "traits": {
    "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
    "smithy.api#required": {}
  }
}
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```



```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            }
        ]
    }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
          "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
              "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",

```

```

        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",

```

```

"members": {
  "ActiveDirectoryIdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
    },
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    }
  }

```

```

    }
  },
  "Products": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
      "smithy.api#required": {}
    }
  },
  "LastStatusCheckDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date of the last status check.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",

```

```

"members": {
  "Username": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IpV4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}):[0-9a-fA-F]{1,4})$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
        }
      ],

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",

```

```

        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",

```

```

        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    }
],
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},

```

```

    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        }
      },

      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Region"
        }
      ]
    }
  ]
},
],
"rules": [
  {
    "conditions": [
      {
        "fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      },
    ],
    "assign": "PartitionResult"
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
        ],

```

true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        },
        {
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "DualStack is enabled but this partition does not support DualStack",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        }
    }

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```



```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
      },
      "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
        }
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```


to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "UserAssociations": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#UserAssociations",
          "traits": {
            "smithy.api#documentation": "<p>The list of user associations for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "UserAssociations": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UserAssociations",
        "traits": {
          "smithy.api#documentation": "<p>The list of user associations for the specified resource.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```


such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

"min": 0,
  "max": 50
},
"smithy.api#sensitive": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```



```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
CreateLicense
Initialize stack step
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
Serialize stack step
  setOperationInput
  ResolveEndpoint
  OperationSerializer
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  Signing
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  OperationDeserializer
  AddTimeOffsetMiddleware
  RecordResponseTiming
  RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //

```

```

// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err
= addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```



```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
    err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

 steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref

```

    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming

```

    RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
    setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
    spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
    AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

```



```

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {

```

```

return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptors(stack, options); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "ListReceivedLicenses",
}
}
{
"smithy": "2.0",
"shapes": {
"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
"type": "integer"
},
"com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
"type": "long"
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
"type": "operation",
"input": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
},
"output": {
"target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
}
}
}
}

```

```

    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
        }
    },
    "traits": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
        "AmiId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
            }
        },
        "InstanceID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
            }
        },
        "InstanceType": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The instance type of the resource.</p>"
            }
        },
        "AccountID": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
            }
        }
    }
},

```

```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see<n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage<n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage<n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    },
    "LastUpdatedTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
      }
    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {

```

```

    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
      },
      "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the

```

configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```
    "type": "boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
  }
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
```

```

        "ref": "UseDualStack"
    },
    true
]
}

],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "aws.partition",
        "argv": [
            {
                "ref": "Region"
            }
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
],

"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "supportsFIPS"
]

```



```

        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}

```

```

    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "rules": [
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

```

```

"properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
]

```

```

        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [

{
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
  "type": "tree"
}

],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ]
}

```

```

        ],
        "type": "tree"
    }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  }
}

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  }
}

```



```

    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

```

```

{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},

{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",
                "pageSize": "MaxResults",
                "items": "Instances"
            }
        }
    }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
<ul>\n
<li>\n
<p>\n
<code>AccountID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>AmiID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>DualSubscription</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceType</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>ProductCode</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Region</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Status</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>UsageOperation</code>\n
</p>\n
</li>\n
</ul>\n
<p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
<ul>\n
<li>\n
<p>\n
<code>contains</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>equals</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Notequal</code>\n
</p>\n
</li>\n
</ul>"
          }
        },
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation":

```

```

"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t\torganization, the returned results will include data aggregated across your accounts in\n\t\tOrganizations.</p>",
      "smithy.api#http": {
        "code": 200,

```

```

    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n        </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>equals</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>Notequal</code>\n            </p>\n            </li>\n        </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  },
},

```

```

    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```



```

"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",

```

```

        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License

```

Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"

```
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]^{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",

```



```

"member": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    }
  }
},
"traits": {
  "smithy.api#length": {
    "min": 1,
    "max": 100
  },
  "smithy.api#uniqueItems": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",

    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "RedHat",
                "value": "RedHat",
                "documentation": "RedHat subscription provider namespace"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",

```

```

        "documentation": "ACTIVE status"
    },
    {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
    },
    {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```



```

    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    },
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {

"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```

)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

```

```

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {

```

```

    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

```

```

}

result, metadata, err := c.invokeOperation(ctx,
"AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```

```

if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.99 aws-sdk-go-v2-service-internal-s3shared

1.19.14

1.99.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package licenssemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.

```

```

Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde

```

```

}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
}

```

```

if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.

```

```

//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware

```

RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

```

```

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

```

```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":

```

```

{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
},
    "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
        "type": "structure",
        "members": {
            "DomainName": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
                }
            },
            "DomainIpv4List": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4List",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                    "smithy.api#length": {
                        "min": 1,
                        "max": 2
                    }
                }
            }
        }
    },

```



```

    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "SELF_MANAGED",
            "value": "SELF_MANAGED",
            "documentation": "SELF_MANAGED type of Active Directory"
          },
          {
            "name": "AWS_MANAGED",
            "value": "AWS_MANAGED",
            "documentation": "AWS_MANAGED type of Active Directory"
          }
        ]
      }
    },
  },

```

```

"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
            }
        },
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {

```

"smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager secret that enables administrators to add or remove users associated with the license server.</p>",

```
    "smithy.api#required": {}
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code> resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials needed for user administration in
```

the Active Directory.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
```

```

    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    }
  }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^([sd]-[0-9a-f]{10})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory

```

```

identity provider for the user.</p>"
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
},

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results

```

from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
            "smithy.api#required": {}
        }
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
        },
        "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}:|[0-9a-fA-F]{1,4}:|:::[0-9a-fA-F]{1,4}){1,6}|[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
      }
    ]
  }

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {

```

```

    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {

```

```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    }
]

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    }
                ]
            }
        ],
    },

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
    },
    true
]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "fn": "getAttr",
                    "argv": [

{
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {

"url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS":
true,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```



```

        "UseDualStack": false
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    },
                    "params": {
                        "UseFIPS": false,
                        "UseDualStack": false,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",

```



```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```



```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
          },
          {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License status.

Status types.LicenseStatus

// License version.

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```



```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

CheckoutLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions


```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```

```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

        "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":

```

```

"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
  "smithy.api#required": {}
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
      }
    }
  },

```



```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
      },
      true
    ]
  }
},
{
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
{
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
{
  "type": "tree"
},
{
  "type": "tree"
},
{

```



```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        }
      ]
    ]
  }
]

```

```

    }

    ],

    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                }
                            ]
                        }
                    ],
                    "supportsFIPS"
                }
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```



```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",

```



```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    }
},
{
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceType</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>ProductCode</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Region</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Status</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>UsageOperation</code>\n                </p>\n                </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>contains</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>equals</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Notequal</code>\n                </p>\n                </li>\n        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\nA token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\n<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional\nobjects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```



```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSources": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```



```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```



```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//

```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```
    //
```

```
    // - Fingerprint
```

```
    //
```

```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```



```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.100 github.com/aws/aws-sdk-go-v2/service/ssooidc 1.35.8

1.100.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//
the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {

```



```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

    }
    if err = addRecursionDetection(stack); err != nil {
        return err
    }
    if err = addRequestIDRetrieverMiddleware(stack); err != nil {
        return err
    }
    if err = addResponseErrorMiddleware(stack); err != nil {
        return err
    }
    if err = addRequestResponseLogging(stack, options);
    err != nil {
        return err
    }
    if err = addDisableHTTPSMiddleware(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutBorrowLicense",
    }
}

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

```

```

//
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```



```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"


```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      }
    }
  },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```



```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```



```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```



```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

"traits": {
  "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
}
},
"traits": {
  "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```



```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Status": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
    "smithy.api#required": {}
  }
},
"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

    },
    "Settings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {

```



```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
// License beneficiary.
```

```
//
```

```
// This member is required.
```

```
Beneficiary *string
```

```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
```

```
//
```

```
//
```

```
This member is required.
```

```
ClientToken *string
```

```
// Configuration for consumption of the license. Choose a provisional
```

```
// configuration for workloads running with continuous connectivity. Choose a
```

```
// borrow configuration for workloads with offline usage.
```

```
//
```

```
// This member is required.
```

```
ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
// License entitlements.
```

```
//
```

```
// This member is required.
```

```
Entitlements []types.Entitlement
```

```
// Home Region for the license.
```

```
//
```

```
// This member is required.
```

```
HomeRegion *string
```

```
// License issuer.
```

```
//
```

```
// This member is required.
```

```
Issuer *types.Issuer
```

```
// License name.
```

```
//
```

```
// This member is required.
```

```
LicenseName *string
```

```
// Product name.
```

```
//
```

```
// This member is required.
```



```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    }
  }
}

```



```

    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
    }
  },
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```

\\n\\t\\tthe same software on your instance.</p>"

```
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \\n\\t\\t\\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```



```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            {
              "fn": "supportsDualStack"
            }
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
            "type": "tree"
        },
        {
            "conditions": [],
            "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
            "type": "error"
        },
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                    },
                    "properties": {},
                    "headers": {}
                },
                {
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```



```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
}

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \\\n\\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  }\n},\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \\\n\\tfield\nisn't empty, there are additional elements that the service hasn't \\\n\\tincluded in this request. Use this token with the\nnext request to retrieve \\\n\\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListRegisteredSubscriptionProviders"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "RegisteredSubscriptionProviders"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```



```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```



```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licenssemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.101 aws-sdk-go-v2-internal-endpoints

2.7.14

1.101.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```



```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

```



```

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```



```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```



```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
        "type": "union",
        "members": {
            "SecretsManagerCredentialsProvider": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            }
        ],
    },

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```



```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        },
        "OwnerAccountId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes an identity provider.</p>"
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
            }
        },
        "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            }
        },
    },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      },
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {

```

```

        "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
        },
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```



```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```



```

        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```



```

    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "for your server.</p><p>The <code>ServerSettings</code> resource contains the settings",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```



```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```



```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},  
middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}  
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addClientRequestID(stack); err != nil {  
    return err  
}  
if err = addComputeContentLength(stack); err != nil {  
    return err  
}  
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addComputePayloadSHA256(stack); err != nil {  
    return err  
}  
if err = addRetry(stack, options); err != nil {  
    return err  
}  
if err = addRawResponseToMetadata(stack); err != nil {  
    return err  
}  
if err = addRecordResponseTiming(stack); err != nil {  
    return err  
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```



```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```



```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

        },
        {
            "conditions": [],
            "error": "FIPS is enabled
but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```



```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```


that you can use to filter the results by your specified criteria. \n \tFor example, you can specify

<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n

<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto streamline results:</p>\n

- \n <p>\n <code>AccountID</code>\n </p>\n \n \n <p>\n <code>AmiID</code>\n </p>\n \n \n <p>\n <code>DualSubscription</code>\n </p>\n \n \n <p>\n <code>InstanceID</code>\n </p>\n \n \n <p>\n <code>InstanceType</code>\n </p>\n \n \n <p>\n <code>ProductCode</code>\n </p>\n \n \n <p>\n <code>Region</code>\n </p>\n \n \n <p>\n <code>Status</code>\n </p>\n \n \n <p>\n <code>UsageOperation</code>\n </p>\n \n \n

<p>For each filter, you can use one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n

- \n <p>\n <code>contains</code>\n </p>\n \n \n <p>\n <code>equals</code>\n </p>\n \n \n <p>\n <code>Notequal</code>\n </p>\n \n

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {

```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```

```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
        {}
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```



```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```



```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.102 github.com/aws/aws-sdk-go-v2/service/internal/presigned-url 1.13.14

1.102.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```



```

)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licenssemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license.


```

//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

```

```

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```



```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
        "DomainName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  }
},
{
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```

```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```



```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```

```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },
    },
    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```

```

        "ref": "UseFIPS"
    },

true

    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },
            {
"fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
    ]
    }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

```

```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
},
"params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```



```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {

```

```

    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

"type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```


empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
]

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```

```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```

```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```

```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```



```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.

```

ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.

//

// This member is required.

Entitlements []types.Entitlement

// Home Region for the license.

//

// This member is required.

HomeRegion *string

// License issuer.

//

// This member is required.

Issuer *types.Issuer

// License name.

//

// This member is required.

LicenseName *string

// Product name.

//

// This member is required.

ProductName *string

// Product SKU.

//

// This member is required.

ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

//

// This member is required.

Validity *types.DatetimeRange

// Information about

the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in

// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde

```

}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

 steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.

```

```

MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {

```

```

    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's  
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {

```

```
"smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration resource specified in the request.</p>"
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
      }
    },
    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
```

```

    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      }
    }
  }

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
        }
    },
    "SubscriptionProviderCreateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
        }
    },
    "SubscriptionProviderUpdateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
        }
    },
    "DualSubscription": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
        }
    },
    "RegisteredWithSubscriptionProvider": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ]
}

```

```

    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
          "type": "boolean"
        },
        "UseFIPS": {

```

```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"

```

```

    },
    "supportsFIPS"
  ]
}
],
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [

        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],

```

```

        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {

```



```

        },
        "type": "endpoint"
    }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            }
        }
    ]
}
}

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```



```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Notequal</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```



```

return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```



```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```

```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.

```

Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```



```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.103 github.com/aws/aws-sdk-go-v2/service/sso 1.30.4

1.103.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```



```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{}  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```

```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```



```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```



```

        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },

```

```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/DeleteLicenseServerEndpoint"

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```

"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        },
        "StatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
            }
        },
        "com.amazonaws.licensemanagerusersubscriptions#InternalServerException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    },
    {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
      "type": "error"
    }
  ],
  {
    "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    }
                ]
            }
        ],
    },

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
  },
  true
]
}
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [

              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
}

```



```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS":
true,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.gov.gov"
    }
},
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```



```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
      "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```



```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```



```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation":
                "<p>Adds tags to a resource.</p>",
            "smithy.api#http": {
                "method": "PUT",
                "uri": "/tags/{ResourceArn}"
            }
        }
    }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
    ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.

```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License status.

Status types.LicenseStatus

// License version.

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read


```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

CheckoutLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```

```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```

```

    "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":

```



```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```

```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
    },
    true
]
}

],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{

```

```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        ]
      }
    ]
  }
]

```

```

    }

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              {
                "ref": "supportsDualStack"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```



```

        "supportsFIPS"
      ]
    },
    true

  ]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    ]
  },
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
      ],
      true
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```



```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceType</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>ProductCode</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Region</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Status</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>UsageOperation</code>\n                </p>\n                </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>contains</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>equals</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Notequal</code>\n                </p>\n                </li>\n        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\nA token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\n<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional\nobjects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSources": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]^{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```



```

    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```

```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//


```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{ }  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```
    //
```

```
    // - Fingerprint
```

```
    //
```



```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```



```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.104 github.com/aws/aws-sdk-go-v2/service/sts 1.41.1

1.104.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//
the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```



```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptors(stack, options); err != nil {
return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

```

```

//
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:  ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop


```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

"DomainName": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
  }
},
"DomainIpv4List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},

"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"

```

```

resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      }
    }
  },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):{1,6}:[0-9a-fA-F]{1,4})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```



```

        "type": "endpoint"
    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",

    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

  "endpoint": {
    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    }
},
"params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
}
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```



```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```



```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/license-server/ListLicenseServerEndpoints"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "LicenseServerEndpoints"
        }
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
                "smithy.api#range": {
                    "min": 1,
                    "max": 100
                }
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```



```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.

```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```


}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```



```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```

```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        }
    },
    "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
    }
},
"Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```

\\n\\t\\tthe same software on your instance.</p>"

```
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \\n\\t\\t\\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```



```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

        "type": "tree"
        },
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                    },
                    "properties": {},
                    "headers": {}
                },
                {
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ]
  }
]

```

```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {

```



```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
  }
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
}
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>Notequal</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  },\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```

```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```



```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```



```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.105 aws-sdk-go-v2-service-sts 1.41.1

1.105.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```
    out := result.(*CheckoutBorrowLicenseOutput)
```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed

```

[]types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return
```

```
    err
```

```
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
```

```
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
```

```
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (

```



```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
```

```

    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.

```

EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.

LicenseArn

*string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack,

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter

```


SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,

```

```

c.addOperationDeleteLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      },
      "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
        "type": "string",
        "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                    ],
                }
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
        }
    ]
}
true

```

```

        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},

```



```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```



```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ],

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```



```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  }
},
"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

```

```

// License status.
Status types.LicenseStatus

```

```

// License version.
Version *string

```

```

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```



```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}
 - name: Checkout this ref
 uses: actions/checkout@v2
 with:
 path: new-ref
 fetch-depth: 0
 - name: Get Diff
 run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
 \${{ github.sha }} > refDiffFiles.txt
 - name: Get Target Files
 run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
 - name: Checkout scancode
 uses: actions/checkout@v2
 with:
 repository: nexB/scancode-toolkit

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```



```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "string"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  }
}

```

```

    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [

```



```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          }
        ]
      }
    ]
  }
]
}

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify `Region` in the `Name`, with the \n \t`contains` operator to list all subscriptions that match a partial string in the \n \t`Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key \n \tto streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to \n \tdefine the behavior of the filter:

- `contains`
- `equals`
- `NotEqual`

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licenseanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licenseanagerlinuxsubscriptions#InstanceList",
      "traits": {

```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```



```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\&#xA0; integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\&#xA0; allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.106 github.com/aws/aws-sdk-go-v2/feature/ec2/imds 1.18.14

1.106.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string
}
```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```



```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license.

```

//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

```

```

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer

```

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The ActiveDirectorySettings resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
        "DomainName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
},
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```

```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```



```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```

```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```

```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    {
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```



```

        "ref": "UseFIPS"
    },
    true
]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}

```

```

        }
    ]
    }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [

```

```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```



```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
},
"params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```

```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```



```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/instance/ListInstances"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "InstanceSummaries"
        }
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {

```

```

    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

"type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
}

```



```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},
"SubscriptionEndDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The end date of a subscription.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```

```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```



```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```

```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
        }
      },
      "UpdateSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
          "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
          "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
              "smithy.api#required": {}
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```

```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
CreateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.

```

ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.

//

// This member is required.

Entitlements []types.Entitlement

// Home Region for the license.

//

// This member is required.

HomeRegion *string

// License issuer.

//

// This member is required.

Issuer *types.Issuer

// License name.

//

// This member is required.

LicenseName *string

// Product name.

//

// This member is required.

ProductName *string

// Product SKU.

//

// This member is required.

ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

//

// This member is required.

Validity *types.DatetimeRange

// Information about

the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in

// License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde


```

}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
 licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

 steps:
 - name: Checkout target
 uses: actions/checkout@v2
 with:
 path: sdkbase
 ref: \${{ github.base_ref }}

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.

```

```

MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {

```

```

    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's  
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {

```



```

    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
        "type": "structure",
        "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
                }
            }
        }
    }

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
        }
    ]
}

```

```

    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {
        "Region": {
          "builtIn": "AWS::Region",
          "required": false,
          "documentation": "The AWS region used to dispatch the request.",
          "type": "string"
        },
        "UseDualStack": {
          "builtIn": "AWS::UseDualStack",
          "required": true,
          "default": false,
          "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
          "type": "boolean"
        },
        "UseFIPS": {

```

```

        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {

```

```

        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    }
                                ]
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

    },
    "supportsFIPS"
  ]
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [

        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [

    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],

```

```

        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",

        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {

```



```

        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
    "type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            }
        }
    ],
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}
]
}
}

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```



```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```


"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  },
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```



```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"
```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html

```

```

func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
    Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.

```


Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack,
```

```
options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.107 [github.com/aws/aws-sdk-go-](https://github.com/aws/aws-sdk-go-v2/service/s3) v2/service/s3 1.91.1

1.107.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckoutBorrowLicenseInput struct {
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    //  
    the request.  
    //  
    // This member is required.  
    ClientToken *string
```

```
    // Digital signature method. The possible value is JSON Web Signature (JWS)  
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].  
    //  
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5  
    //  
    // This member is required.  
    DigitalSignatureMethod types.DigitalSignatureMethod
```

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```



```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)  
  
// Checks out the specified license.  
//  
// If the account that created the license is the same that is performing the  
// check out, you must specify the account as the beneficiary.  
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))  
(*CheckoutLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutLicenseInput{  
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,  
c.addOperationCheckoutLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
  
    out := result.(*CheckoutLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}  
  
type CheckoutLicenseInput struct {  
  
    //  
    Checkout type.  
    //  
    // This member is required.  
    CheckoutType types.CheckoutType  
  
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.  
    //  
    // This member is required.  
    ClientToken *string  
  
    // License entitlements.  
    //
```

```

// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

```



```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion

```

```

*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```



```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains
details about the Active Directory, including network access details such as domain name and IP addresses, and the
credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```

href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },

```

```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```



```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

    "uri": "/license-server/DeleteLicenseServerEndpoint"

  },
  "smithy.api#idempotent": {}
}

},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}

},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([sd]-[0-9a-f]{10})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```

```

"input": {

"target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```

"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```



```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [

```



```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    }
                ]
            }
        ],
    },

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
    },
    true
]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "fn": "getAttr",
                    "argv": [

{
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
    }
],
"rules": [
{
    "conditions": [],
    "endpoint": {

"url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
}

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS":
true,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":

```



```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
},
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":

```

```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```



```

    }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/ListIdentityProviders"
        },
        "smithy.api#paginated": {
            "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
}
},

```

```

    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```

```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```



```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  }
}

```



```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

        "smithy.api#httpError": 404
    }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
        "SecretId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                "smithy.api#length": {
                    "min": 1
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```



```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
    ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```



```
// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
```

Validity *types.DatetimeRange

// Information about
the license.

LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
License Manager, see the [TagResource]operation.

//

// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html

Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

```

CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
      # compare
      - name: License test
        run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses

```

are the same. Success."; exit 0; fi
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
AssociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

```

Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //

```

```

// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}

```

```

    }
  }
  {
    "smithy": "2.0",
    "shapes": {
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
        "type": "integer"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
        "type": "long"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
        },
        "output": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
          },
          {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
          }
        ],
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
      "type": "structure",
      "members": {

```

```

    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
      "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
      },
      "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
          "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {

```



```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":

```



```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetServiceSettings"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {

```

```

"type": "structure",
"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    }
  },

```

```

    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",
    "smithy.rules#endpointRuleSet": {
      "version": "1.0",
      "parameters": {

```



```

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},
"UseFIPS": {
  "builtIn": "AWS::UseFIPS",
  "required": true,
  "default": false,
  "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
  "type": "boolean"
},
"Endpoint": {
  "builtIn": "SDK::Endpoint",
  "required": false,
  "documentation": "Override the endpoint used to send this request",
  "type": "string"
}
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {

```

```

        "ref": "UseFIPS"
    },
    true
]
}

],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{

```

```

"conditions": [],
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [

"fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {

"ref": "UseFIPS"
            },
            true
          ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          ]
        }
      ]
    ]
  }
]

```

```

    }

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ],
        "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-

```

```

fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
},
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ],
            }
          ]
        }
      ]
    }
  ]
}

```

```

        "supportsFIPS"
      ]
    },
    true

  ]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
        }
      }
    ],
    "headers": {},
    "type": "endpoint"
  },
  {
    "type": "tree"
  },
  {
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
  },
  {
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseDualStack"
          },
          true
        ]
      }
    ]
  }
]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    "params": {

```



```

        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {

```

```

        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {

"Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",

```

```

        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>AccountID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>AmiID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>DualSubscription</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceID</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>InstanceType</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>ProductCode</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Region</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Status</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>UsageOperation</code>\n                </p>\n                </li>\n        </ul>\n        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \tdefine the behavior of the filter:</p>\n        <ul>\n            <li>\n                <p>\n                    <code>contains</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>equals</code>\n                </p>\n                </li>\n            <li>\n                <p>\n                    <code>Notequal</code>\n                </p>\n                </li>\n        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\nA token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  }
},
"traits": {
  "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "\n<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional\nobjects.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Subscriptions"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    <p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n    <code>contains</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>equals</code>\n    <p>\n    </li>\n    <li>\n    <p>\n    <code>Notequal</code>\n    <p>\n    </li>\n    </ul>"
        }
      },
    },
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
      "type": "operation",
      "input": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"

```

```

    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSources": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",

```



```

    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      },
      "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",

```

```

        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
    },
    "Type": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
    },
    "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
        "type": "list",

        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
            "type": "string",
            "traits": {
                "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
            "type": "string",
            "traits": {
                "smithy.api#enum": [
                    {
                        "name": "RedHat",
                        "value": "RedHat",
                        "documentation": "RedHat subscription provider namespace"
                    }
                ]
            }
        }
    }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {

```

```

    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to which to add the specified \n\t\t\t\t\tmetadata tags.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource. Tags are \n\t\t\t\t\tformatted as key value pairs.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":

```

```

    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {

```

```

        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

// The type of license to remove from the workspace.

//

// This member is required.

LicenseType types.LicenseType

//

The ID of the workspace to remove the Grafana Enterprise license from.

//

```

// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{ }, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,  
c.addOperationListLicensesMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
}
```

```
    out := result.(*ListLicensesOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type ListLicensesInput struct {
```

```
    // Filters to scope the results. The following filters are supported:
```

```
    //
```

```
    // - Beneficiary
```

```
    //
```

```
    // - ProductSKU
```

```
    //
```

```
    // - Fingerprint
```

```
    //
```

```

// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"

```



```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.108 aws-sdk-go-v2-service-internal-presigned-url 1.13.14

1.108.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```



```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//
the request.
//
// This member is required.
ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

```



```

//
Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.

```

```

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```

```

return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```



```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckoutBorrowLicense
 Initialize stack step
 RegisterServiceMetadata
 legacyEndpointContextSetter
 SetLogger
 OperationInputValidation
 Serialize stack step
 setOperationInput
 ResolveEndpoint
 OperationSerializer
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 Signing
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 OperationDeserializer
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory  either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```



```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```



```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4})::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```



```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",

    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

  "endpoint": {
    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
  }
}

```



```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        }
    }

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```



```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
  },

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```



```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      }
    },
    "IdentityProvider":

```



```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A parameter is not valid.</p>",
    "smithy.api#error": "client"
  }
}
}
}

```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop


```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)

```

```
out.ResultMetadata = metadata
return out, nil
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```

```

    ProductName *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // Date and time range during which the license is valid, in ISO8601-UTC format.
    //
    // This member is required.
    Validity *types.DatetimeRange

    // Information about
    the license.
    LicenseMetadata []types.Metadata

    // Tags to add to the license. For more information about tagging support in
    // License Manager, see the [TagResource]operation.
    //
    // [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
    Tags []types.Tag

    noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

}

```
func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {  
    return &awsmiddleware.RegisterServiceMetadata{  
        Region:    region,  
        ServiceID: ServiceID,  
        OperationName: "CreateLicense",  
    }  
}
```

CheckInLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

OperationDeserializer

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

name: License Scan

on: [pull_request]

permissions:

contents: read

jobs:

licensescan:

name: License Scan

runs-on: ubuntu-latest

strategy:

matrix:

python-version: [3.9]

steps:

- name: Checkout target

uses: actions/checkout@v2

with:

path: sdkbase

ref: \${ github.base_ref }

- name: Checkout this ref

uses: actions/checkout@v2

with:

path: new-ref

fetch-depth: 0

- name: Get Diff

run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${ github.event.pull_request.base.sha }

\${ github.sha } > refDiffFiles.txt

- name: Get Target Files

run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

- name: Checkout scancode

uses: actions/checkout@v2

with:

repository: nexB/scancode-toolkit

path: scancode-toolkit

fetch-depth: 1

- name: Set up Python

\${ matrix.python-version }

uses: actions/setup-python@v2

with:

python-version: \${ matrix.python-version }

ScanCode

- name: Self-configure scancode

working-directory: ./scancode-toolkit

run: ./scancode --help

```

- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

Signing

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
                }
            ],
            "traits": {
                "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
                "smithy.api#http": {
                    "code": 200,

```



```

        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        }
    },
    "Operator": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    }
  }
}

```

```

    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a href='\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-operation.html\">Usage</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for

```

\\n\\t\\tthe same software on your instance.</p>"

```
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \\n\\t\\t\\ta
third-party Linux subscription provider that you've registered with License Manager.</p>"
```

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
```

```
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",

```

```

        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}

```

```

    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "aws.partition",
              "argv": [
                {
                  "ref": "Region"
                }
              ],
              "assign": "PartitionResult"
            }
          ],
          "rules": [
            {
              "conditions": [
                {
                  "fn": "booleanEquals",
                  "argv": [
                    {
                      "ref": "UseFIPS"
                    },
                    true
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],

```

```

        "rules": [
            {
                "conditions": [],

"rules": [

                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],

            "type": "tree"
        },
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    },
    "rules": [
        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
"type": "tree"
},
{

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }

```



```

        "type": "tree"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  }
},
{
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},

```

```

{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  }
}

```

```

        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    }
  },

```



```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>DualSubscription</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceID</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>InstanceType</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>ProductCode</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Region</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>Status</code>\n
    </p>\n
    </li>\n
  <li>\n
    <p>\n
      <code>UsageOperation</code>\n
    </p>\n
    </li>\n
</ul>\n
<p>For each filter, you can use

```

```

one of the following Operator values to \n \t\tdefine the behavior of the filter:</p>\n      <ul>\n
    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
        returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Subscriptions"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

          "traits": {
            "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
            that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
            <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't

```

```

match a certain subscription's name.</p>\n      <p>The valid names for this filter are:</p>\n      <ul>\n        <li>\n          <p>\n            <code>Subscription</code>\n          </p>\n        </li>\n      </ul>\n    <p>The valid Operators for this filter are:</p>\n    <ul>\n      <li>\n        <p>\n          <code>contains</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>>equals</code>\n        </p>\n      </li>\n      <li>\n        <p>\n          <code>NotEqual</code>\n        </p>\n      </li>\n    </ul>\n  }\n},\n  \"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n  },\n  \"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n      \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the\nnextToken from a previously truncated response.</p>\",\n      \"smithy.api#length\": {\n        \"min\": 1,\n        \"max\": 16384\n      }\n    }\n  }\n},\n  \"traits\": {\n    \"smithy.api#documentation\": \"NextToken\nlength limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n  }\n},\n  \"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n      \"Subscriptions\": {\n        \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>An array that contains subscription objects.</p>\"\n        }\n      },\n      \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n          \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield\nisn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the\nnext request to retrieve \n\tadditional objects.</p>\"\n        }\n      }\n    }\n  }\n}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",

```



```

        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```



```

    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
    },
    "LastSuccessfulDataRetrievalTime": {
        "target": "smithy.api#String",
        "traits": {

            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",

```

```

"traits": {
  "smithy.api#enum": [
    {
      "name": "InProgress",
      "value": "InProgress",
      "documentation": "InProgress status"
    },
    {
      "name": "Completed",
      "value": "Completed",
      "documentation": "Completed status"
    },
    {
      "name": "Successful",
      "value": "Successful",
      "documentation": "Successful status"
    },
    {
      "name": "Failed",
      "value": "Failed",
      "documentation": "Failed status"
    }
  ]
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  }
}

```

```

    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },

```



```

    }
  },
  "tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
            "smithy.api#httpQuery": "tagKeys",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "LinuxSubscriptionsDiscovery": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
    "traits": {
      "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
      "smithy.api#required":
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Removes the Grafana Enterprise license from a workspace.
```

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,  
c.addOperationDisassociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*DisassociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type DisassociateLicenseInput struct {
```

```

// The type of license to remove from the workspace.
//
// This member is required.
LicenseType types.LicenseType

//
The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```



```

    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}

```

Apache License
 Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (
```

```
    "context"
```

```
    "fmt"
```

```
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
```

```
    "github.com/aws/smithy-go/middleware"
```

```
    smithyhttp "github.com/aws/smithy-go/transport/http"
```

```
)
```

```
// Lists the licenses for your account.
```

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
```

```
(*ListLicensesOutput, error) {
```

```
    if params == nil {
```

```
        params = &ListLicensesInput{}
```

```
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
```

```
    c.addOperationListLicensesMiddlewares)
```

```
    if err != nil {
```

```
        return nil, err
```

```
    }
```

```
    out := result.(*ListLicensesOutput)
```

```
    out.ResultMetadata = metadata
```

```
    return out, nil
```

```
}
```

```
type ListLicensesInput struct {
```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```



```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licenseseanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

1.109 aws-sdk-go-v2-internal-v4a 1.4.14

1.109.1 Available under license :

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```
    out := result.(*CheckoutBorrowLicenseOutput)
```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed

```

[]types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return
```

```
    err
```

```
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
```

```
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
```

```
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutBorrowLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (

```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
```

```

    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}

```



```

    }
    if err = addInterceptAttempt(stack, options); err != nil {
        return err
    }
    if err = addInterceptors(stack, options); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "GetLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.

```

EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.

LicenseArn

*string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack,

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}
DisassociateLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter

```

SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,

```

```

c.addOperationDeleteLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```



```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

```

{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {
              "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ],

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
}

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```



```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          },
          },
          "traits": {
            "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
          "type": "list",
          "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
          }
        },
        "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
          "type": "union",
          "members": {
            "ActiveDirectoryIdentityProvider": {
              "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
              "traits": {
                "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
              }
            }
          },
          },
          "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      },
      "FailureMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        },
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                    "smithy.api#required": {}
                }
            },
            "Products": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
                "traits": {
                    "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                    "smithy.api#required": {}
                }
            },
        },
    },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {

```



```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                },
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        }
                    ],
                    "value": true
                }
            ],
            "assign": "UseFIPS"
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                }
            ],
            "value": true
        }
    ]
}

```

```

        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    },
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        ]
    }
]

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```



```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```



```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```



```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
        },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```


or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
    "smithy.api#required": {}
  },
  "RemoveSubnets": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}
```

CreateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {

```

```

if params == nil {
    params = &CreateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //

```

```

// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

```

```

// License status.
Status types.LicenseStatus

```

```

// License version.
Version *string

```

```

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},  
middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}  
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addClientRequestID(stack); err != nil {  
    return err  
}  
if err = addComputeContentLength(stack); err != nil {  
    return err  
}  
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}  
if err = addComputePayloadSHA256(stack); err != nil {  
    return err  
}  
if err = addRetry(stack, options); err != nil {  
    return err  
}  
if err = addRawResponseToMetadata(stack); err != nil {  
    return err  
}  
if err = addRecordResponseTiming(stack); err != nil {  
    return err  
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}
CheckInLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
name: License Scan
runs-on: ubuntu-latest
strategy:
matrix:
python-version: [3.9]

steps:
- name: Checkout target
uses: actions/checkout@v2
with:
path: sdkbase
ref: \${{ github.base_ref }}
- name: Checkout this ref
uses: actions/checkout@v2
with:
path: new-ref
fetch-depth: 0
- name: Get Diff
run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
\${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
uses: actions/checkout@v2
with:
repository: nexB/scancode-toolkit


```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

AssociateLicense

Initialize stack step

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

Serialize stack step

setOperationInput

ResolveEndpoint

OperationSerializer

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
CheckoutLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
ListReceivedLicenses
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

GetLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput
ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
DeleteLicense
Initialize stack step
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
Serialize stack step
setOperationInput

```

ResolveEndpoint
OperationSerializer
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
Signing
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
OperationDeserializer
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {

```

```

if params == nil {
    params = &ListReceivedLicensesInput{}
}

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}

```



```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
{
    "smithy": "2.0",
    "shapes": {
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
            "type": "integer"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
            "type": "long"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
                },
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
                }
            ]
        }
    }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id"
],
"additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "string"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "string"
        }
    }
}

```

```

    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```

```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
},

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```



```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    }
},
{
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```



```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify `Region` in the `Name`, with the `contains` operator to list all subscriptions that match a partial string in the `Value`, such as `us-west`.

For each filter, you can specify one of the following values for the `Name` key \n \tto streamline results:

- `AccountID`
- `AmiID`
- `DualSubscription`
- `InstanceID`
- `InstanceType`
- `ProductCode`
- `Region`
- `Status`
- `UsageOperation`

For each filter, you can use one of the following `Operator` values to \n \tdefine the behavior of the filter:

- `contains`
- `equals`
- `Notequal`

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {

```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {

```



```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/UpdateServiceSettings"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
{}
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Removes the Grafana Enterprise license from a workspace.

```
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns  
...func(*Options)) (*DisassociateLicenseOutput, error) {  
    if params == nil {  
        params = &DisassociateLicenseInput{  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```



```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DisassociateLicense",
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Lists the licenses for your account.

```
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))  
(*ListLicensesOutput, error) {  
    if params == nil {  
        params = &ListLicensesInput{}  
    }  
}
```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```



```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack,
options); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```



```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptors(stack, options); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckInLicense",
}
}

```

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2025 Cisco Systems, Inc. All rights reserved.